Comments on “Structure-driven Multiple Constraint Acquisition”

Dimosthenis C. Tsouros[1], Kostas Stergiou[1], Christian Bessiere[2]

[1] Dept. of Informatics & Telecommunications Engineering,
University of Western Macedonia, Kozani, Greece
dtsouros@uowm.gr, kstergiou@uowm.gr
[2] CNRS, University of Montpellier, France
bessiere@lirmm.fr

Abstract. In the paper “Structure-driven Multiple Constraint Acquisition”, presented at CP-2019, we detected two important drawbacks of multiple constraint acquisition, and proposed two techniques (integrated into a new algorithm) to overcome them. Specifically, we proposed a constraint acquisition algorithm that exploits the structure of the learned constraint network to focus the interactive learning process on more promising points of the problem, while it avoids searching exhaustively to learn all the constraints from the generated example, taking a more simple approach. In this paper we recall some details about the development of the new algorithm.

1 Background on Constraint Acquisition

The basic assumption in CP is that the user models the problem and a solver is then used to solve it. However, modeling is considered as a major bottleneck in the wider use of constraint technology [1–3], as it requires considerable expertise in CP. To overcome this obstacle, several techniques have been proposed [4–9]. State-of-the-art interactive constraint acquisition systems such as QuAcq [10] and MultiAcq [11] can assist non-expert users in the modelling task. The main idea is that a series of examples/queries is posted to the user, and the model of the target constraint problem is acquired (i.e. learned) based on the answers of the user. In more detail, the system tries to identify if a constraint belongs to the target network by generating a variable assignment that violates it. Then, this assignment is posted to the user asking whether the assignment is a solution to the problem or not, and based on the answer of the user, the constraint can either be added to the learned network or removed from the candidate constraints.

QuAcq learns one constraint from each generated query, while MultiAcq learns all the constraints that can possibly be learned. However, MultiAcq needs a linear number of queries to find the scope of each constraint, while QuAcq has a logarithmic complexity in terms of the number of queries.

Despite the progress being made in constraint acquisition, there are still important challenges to be faced regarding the applicability of the existing methods and their computational cost. MQuAcq [12] is an algorithm for active constraint acquisition that has been shown to outperform previous algorithms such as QuAcq and MultiAcq. It combines the strengths of QuAcq and MultiAcq and outperforms both of them, as it requires a logarithmic number of queries to locate the scope of each violated constraint, and discovers all the violated constraints from a negative example. This is done via a recursive process, by removing one variable from the scope of each constraint learned, until it finds a query that does not violate any constraint already found.

In our CP-2019 paper “Structure-driven Multiple Constraint Acquisition” we displayed two important drawbacks of MQuAcq [13]. First, for each negative example, the number of recursive calls to the main procedure of MQuAcq can be non-linear, making it impractical in terms of cpu time for large problems. Second, MQuAcq, as well as QuAcq and MultiAcq, does not take into account the structure of the learned problem. In the next section we recall how we discovered these flaws of MQuAcq and developed the new algorithm.

2 Developing “Structure-driven Multiple Constraint Acquisition”

Although it is not always easy to pinpoint when and how the ideas forming the basis of a paper came about, we can safely say that our paper was developed as the result of the combination of three factors:

1. A natural evolution of the work on MQuAcq presented at CP-2018.
2. Interaction with fellow researchers during CP-2018.
3. Questions arising through unexpected experimental results.

Regarding the first of the above, exploiting the structure of the problem that is being acquired to better focus the generated queries seemed like a natural step to take while trying to improve the performance of MQuAcq. So this was a goal that was set after our work on MQuAcq was completed.

Regarding the second factor, after the presentation of our paper on MQuAcq at CP-2018 by Dimosthenis Tsouros, among the questions posted to him, one fellow researcher inquired about the possibility of exploiting structure during the acquisition process, while another asked whether the time complexity of MQuAcq had been specified. The first question helped to confirm our intuition about the importance of problem structure, which was the basis of one of the two main contributions of our CP-2019 paper. The second question was discussed between us after CP-2018 but was rather cast aside to focus on other things as it did not seem very important to us at that time.

However, the importance of MQuAcq’s (then unknown) time complexity emerged as a serious issue some months later when, out of scientific curiosity, we decided to test the existing constraint acquisition algorithms on larger problems than the ones considered in the literature. Unexpectedly to us, experimental results on large problems (like 16×16 Sudoku) showed that MQuAcq, which was considerably faster than its predecessor QuAcq on smaller problems, was now quite slower. This forced us to revisit the question about MQuAcq’s complexity. We found that the number of iterations performed by the algorithm's main procedure to find all the violated constraints from a negative example was non-linear in the worst case, and this seriously affects its run time as the size of the problems grows.

Based on these, we developed MQuAcq-2, an algorithm that instead of finding *all* the violated constraints from each negative example, it tries to focus on the most promising parts derived from the structure of the learned network. To avoid the non-linear time-complexity of MQuAcq, we used a "lazy" approach by removing the entire scope of each learned constraint from the next query, so that all the learned constraints from a single example are non-overlapping. To be more precise, the main contributions of MQuAcq-2 are the following:

- It exploits the structure of the learned network to focus on selected violated constraints. Specifically, we presented an instantiation of the general method where the structure searched for is that of a quasi-clique, as cliques and quasi-cliques are common in CSPs, but the algorithm could also exploit other types of structure.
- In case no more constraints can be learned by exploiting the structure of the learned network, it tries to find some non-overlapping constraints of the target network. Hence, it can learn some of the violating constraints but not necessarily all of them. This allows us to alleviate the high run time that MQuAcq incurs when searching for all the violated constraints from each negative example.

The combination of these two contributions resulted in an algorithm that outperforms MQuAcq (as well as QuAcq and MultiAcq) in terms of both time and number of queries, especially on larger problems, and even in the absence of structure.

3 Conclusions

The ideas and motivation behind "Structure-driven Multiple Constraint Acquisition" came about not only through a will to improve our existing algorithm, but also through interaction with other researchers at CP, and through unexpected experimental results. This demonstrates the ever lasting usefulness of conference presentations and interaction with our peers, and the importance of curiosity-driven experimentation.

References

1. Freuder, E.C.: Modeling: the final frontier. In: The First International Conference on The Practical Application of Constraint Technologies and Logic Programming (PACLP), London. (1999) 15–21
2. Freuder, E.C., O'Sullivan, B.: Grand challenges for constraint programming. Constraints **19**(2) (2014) 150–162

3. Freuder, E.C.: Progress towards the holy grail. Constraints **23**(2) (2018) 158–171
4. Freuder, E.C., Wallace, R.J.: Suggestion strategies for constraint-based matchmaker agents. In: International Conference on Principles and Practice of Constraint Programming, Springer (1998) 192–204
5. Bessiere, C., Coletta, R., Freuder, E.C., O'Sullivan, B.: Leveraging the learning power of examples in automated constraint acquisition. In: International Conference on Principles and Practice of Constraint Programming, Springer (2004) 123–137
6. Bessiere, C., Coletta, R., Koriche, F., O'Sullivan, B.: A sat-based version space algorithm for acquiring constraint satisfaction problems. In: European Conference on Machine Learning, Springer (2005) 23–34
7. Lallouet, A., Lopez, M., Martin, L., Vrain, C.: On learning constraint problems. In: 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI). Volume 1., IEEE (2010) 45–52
8. Beldiceanu, N., Simonis, H.: A model seeker: Extracting global constraint models from positive examples. In: International Conference on Principles and Practice of Constraint Programming, Springer (2012) 141–157
9. Bessiere, C., Koriche, F., Lazaar, N., O'Sullivan, B.: Constraint acquisition. Artificial Intelligence **244** (2017) 315–342
10. Bessiere, C., Coletta, R., Hebrard, E., Katsirelos, G., Lazaar, N., Narodytska, N., Quimper, C.G., Walsh, T., et al.: Constraint acquisition via partial queries. In: International Joint Conference on Artificial Intelligence (IJCAI). Volume 13. (2013) 475–481
11. Arcangioli, R., Bessiere, C., Lazaar, N.: Multiple constraint aquisition. In: International Joint Conference on Artificial Intelligence (IJCAI). (2016) 698–704
12. Tsouros, D.C., Stergiou, K., Sarigiannidis, P.G.: Efficient methods for constraint acquisition. In: 24th International Conference on Principles and Practice of Constraint Programming. (2018)
13. Tsouros, D.C., Stergiou, K., Bessiere, C.: Structure-driven multiple constraint acquisition. In: 25th International Conference on Principles and Practice of Constraint Programming. (2019)