Efficient Methods for Constraint Acquisition

Dimosthenis C. Tsouros

Dept. of Informatics & Telecommunications Engineering,
University of Western Macedonia, Kozani, Greece
dtsouros@uowm.gr

Abstract. 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, as it requires considerable expertise in CP. To overcome this obstacle, several techniques have been proposed. State-of-the-art interactive constraint acquisition systems such as QuAcq and MultiAcq 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. 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. In this PhD dissertation I intend to address some of these challenges. Specifically, the main goals are: 1) To develop novel techniques for constraint acquisition that improve the performance of the systems in terms of queries required and cpu time, 2) To make constraint acquisition applicable by lifting some of the restricting assumptions that current methods make. So far, I have dealt with the first goal, having introduced new methods that can boost the performance of constraint acquisition systems. I have presented an algorithm that learns several constraints from each generated example, decreasing the total time needed for the acquisition process. In addition, I developed novel heuristics that can be applied during query generation to boost the performance of constraint acquisition algorithms. Finally, I developed an improved algorithm that further decreases the total time of the acquisition process in large problems and also a technique to exploit the structure of the problem to better focus the queries to the user.

Keywords: constraint acquisition, learning, modeling

1 Introduction

Constraint programming (CP) has made significant progress over the last decades, and is now considered as one of the foremost paradigms for solving combinatorial

Dimosthenis C. Tsouros is the PhD Student
Kostas Stergiou is the Supervisor
Panagiotis G. Sarigiannidis and Christian Bessiere are the members of the PhD committee

problems. The basic assumption in CP is that the user models the problem and a solver is then used to solve it. Despite the many successful applications of CP on combinatorial problems from various domains, there are still challenges to be faced in order to make CP technology even more widely used. One of the major challenges in CP research is that of efficiently obtaining a good model of a real problem without relying on expert human modellers [1–3]. Hence, a number of approaches to automated modeling have been developed using Machine Learning methods [4, 5].

Along these lines, an area of research that has started to attract a lot of attention is that of *constraint acquisition* where the model of a constraint problem is acquired (i.e. learned) using a set of examples that are posted to a human user or to a software system [6]. Constraint acquisition is an area where constraint programming meets machine learning, as the problem can be formulated as a concept learning task. In particular, inductive learning methods are employed to learn the model of a problem using a set of examples that are posted to a user.

Constraint acquisition can come in various flavors depending on factors such as whether the learner can post queries to the user dynamically, and the type of queries that can be posted and answered by the user. In *passive* acquisition, examples of solutions and non-solutions are provided by the user. Based on these examples, the system learns a set of constraints that correctly classifies all the given examples [7–10, 6]. A major limitation of passive acquisition (and passive learning in general) is the requirement, from the user's part, to provide diverse examples of solutions and non-solution to the system.

In contrast, *active* or *interactive* acquisition systems interact with the user while acquiring the constraint network. This is a special case of query-directed learning, also known as "exact learning" [11, 12]. In such systems, the basic query is to ask the user to classify an example as solution or not solution. This "yes/no" type of question is called membership query [13], and this is the type of query that has received the most attention in active constraint acquisition [14, 6, 15]. The system can also ask the user to classify partial examples [16] or to provide a violated constraint when a proposed example is considered as incorrect [17]. Other types of queries, e.g. recommendation and generalization ones, have also been considered [18, 19], but answering such queries requires a higher level of expertise from the user's part.

A state-of-the-art interactive acquisition algorithm is QuAcq [16]. QuAcq is able to ask the user to classify partial queries, which may be easier for the user to answer. Also, asking partial queries gives the system the capability to focus on the scope of a constraint that is violated and hence learn the constraint. If the answer to a membership query posted by QuAcq is positive, the system reduces the search space by removing the set of constraints violated by this example. If the answer is negative, QuAcq asks a series of partial queries to locate the scope of one of the violated constraints of the target network and then learn the constraint. QuAcq needs a number of queries logarithmic in the size of the example to locate the scope of a violated constraint. Another relevant algorithm is MultiAcq [20]. This algorithm learns all the constraints of the target network that are violated by a negative example, but it needs a linear number of queries to learn each one.

2 Research Questions

Active acquisition decreases significantly the number of examples necessary to converge to the target set of constraints. In addition, it does not require the existence of diverse examples of solutions and non-solutions to the problem. This is an important advantage especially if the problem has not already been solved, and so no solution can be provided. Also, it is not required that the user is human, as it could be a previous system developed to solve the problem [21]. However, active learning still presents important challenges in order to be applicable in real problems.

One such challenge is the high computational cost of the existing methods in terms of the number of queries required and the cpu time needed to converge to the target network. Algorithms such as QuAcq can be unacceptably slow and require an unrealistic number of queries even on small problems.

Another important challenge is the restrictive assumptions made by existing systems. For instance, it is assumed that the user can always reply to any given query and is always correct in his/her replies. This is clearly not the case in practice.

The resulting research questions are the following:

- Can new strategies be devised in order to decrease the total time of the acquisition process?
  - How can the system generate useful queries faster?
  - How can the system acquire more information from each generated example faster?
- How can the number of queries needed to converge to the target constraint network be decreased?
  - How can the system generate more informative queries in reasonable time?
  - How can the system acquire more information from each generated example?
  - How can the system avoid posting redundant queries to the user?
  - What kind of queries can be exploited to boost the acquisition process?
  - Can the system exploit the structure of the problem to focus the queries posted?
- Can a constraint acquisition system learn a constraint network when there are omission and errors in the answers of the user?
- Can all the developed methods be integrated in a software tool that can assist real users to model real problems?

The aim of the PhD dissertation is to develop and implement novel techniques for learning constraint satisfaction problems that deal with the above mentioned issues. To be more precise, the goals are:

1. To improve the performance of constraint acquisition systems from a computational aspect, regarding mainly the total time of the acquisition process and the number of queries needed. This includes:
   – The development of novel constraint acquisition algorithms that acquire more information from each generated example, to decrease the number of queries needed and the total time of the acquisition process.
   – The development of techniques to exploit the structure of the learned network to better focus the queries.
   – The optimization of the process that finds the scopes of the constraints, by avoiding redundant queries.
   – The improvement of the query generation process, via new heuristics and techniques, to generate more informative queries in reasonable time.
2. To extend the current methods and develop new ones that can make constraint acquisition applicable in real-world scenarios. Specifically, I intend to:
   – Develop methods to deal with cases in which omissions and/or errors in the answers of the user are possible.
   – Integrate all the developed techniques in a software tool that can assist non-expert modelers in their task.

3 Progress so far

In this direction, I developed methods that deal with a lot of the above issues, having almost completed the first main goal, i.e. to improve the performance of constraint acquisition systems. So far, I have developed an algorithm, called MQuAcq, that blends the main idea of MultiAcq into QuAcq resulting in a method that learns as many constraints as MultiAcq does after a negative example, but with a lower complexity [22]. In addition, I proposed a technique that helps reduce the number of queries significantly, while trying to locate the scope of a violated constraint, specifically in the function *FindScope* [22]. Then, I focused on the query generation process and proposed novel heuristics that can be applied during query generation to boost the performance of constraint acquisition algorithms. The above are included in a paper submitted to the Constraints journal which is under the second round of review. Finally, I developed an algorithm that decreases the total time of the acquisition process in large problems and also a technique to exploit the structure of the problem to better focus the queries to the user [23].

3.1 MQuAcq

I designed an algorithm, called MQuAcq, that blends the main idea of MultiAcq into QuAcq, achieving a better complexity bound than MultiAcq. This algorithm uses the reasoning of QuAcq when searching for constraints to learn once a negative query is encountered, but instead of focusing on one constraint, it learns a maximum number of constraints, just like MultiAcq does. But whereas MultiAcq learns constraints of the target network in a number of queries linear in the size of the example, my proposed approach finds constraints in a logarithmic number of queries.

The main difference between QuAcq and MQuAcq is the fact that QuAcq finds one explanation (constraint) of why the user classified an example as negative, whereas MQuAcq learns all the violated constraints. This is done by calling function *FindScope* iteratively, while reducing the search space by removing variables from the scopes of the constraints already found. The main difference between MQuAcq and MultiAcq is that the former uses the QuAcq search method to find each scope through function *FindScope*, and in this way avoids some redundant searches (which can be very time-consuming) as well as queries that MultiAcq makes with its method.

Experiments demonstrate that MQuAcq outperforms both QuAcq and MultiAcq in all important metrics, i.e. total time of the acquisition process, average waiting time for the user between two queries, total number of queries.

3.2 Optimizing the process of locating the scopes of the constraints

I developed an optimization on the process of locating scopes that reduces the number of queries needed to learn the target constraint network significantly. This is done by using the bias (i.e. "candidate" constraints for the problem) to avoid posting redundant queries to the user. As the system knows the violated constraints from the bias before it posts each query to the user, it can sometimes use this information and previous answers to classify the example as negative or positive without new information by the user.

3.3 Query Generation

The next step was to focus on the query generation process, which is a very important step of constraint acquisition that has not been discussed in detail in the literature. More specific:

- I have developed a query generator
- I proposed novel heuristics that can be applied during query generation to boost the performance of constraint acquisition algorithms

In more detail, first I proposed a heuristic that generalizes the idea of allowing partial queries to be posted to the user. Instead of using partial queries only when trying to focus on one or more constraints after a complete example has been classified as negative, I allow the generation of partial examples to be posted as partial queries to the user. As experiments demonstrate, this can reduce the time needed for the system to converge, resulting in avoidance of premature convergence and reduced total run time for the acquisition process.

In addition, I developed variable and value ordering heuristics for the query generation process, aiming at generating queries with more information, and achieving to reduce the maximum cpu time needed for query generation. This is done by exploiting the information from the bias.

3.4 Exploiting the structure of the problem

I further enhanced the efficiency of active constraint acquisition algorithms by proposing a technique for exploiting the structure of the problem while learning the target constraint network [23]. The type of structure that I have investigated so far is that of tightly connected groups of variables that form *quasi-cliques* that are being revealed during the acquisition process. Quasi-cliques are sub-graphs with an edge density exceeding a threshold parameter. The quasi-cliques found could be extendable to complete cliques, so the technique developed focuses on constraints that could extend them.

The proposed algorithm, MQuAcq-2, also alleviates the high cpu time requirements of MQuAcq by acquiring multiple constraints from each generated negative example, but not trying to learn all of them by exhaustively searching in the generated example. Experimental results with benchmark problems demonstrate that MQuAcq-2 offers significant improvements compared, both in terms of time and number of queries, especially on large problems. Importantly, the new algorithm outperforms previous systems even in the absence of structure.

4 Conclusions and Future Work

A major bottleneck in the use of CP is modeling. Expressing a combinatorial problem as a constraint network requires considerable expertise in the field. Constraint acquisition has started to receive increasing attention as a useful tool for automated problem modeling in CP. As a result, a number of acquisition algorithms have been proposed, with QuAcq and MultiAcq being prime examples. However, two bottlenecks of such algorithms are the large number of queries required to converge to the target network, and the high cpu times needed to generate queries, especially near convergence.

In this PhD dissertation two main goals have been set: 1) to develop novel techniques for constraint acquisition that improve its performance, 2) Develop methods that can make constraint acquisition applicable in real situations. So far, I have dealt with the first goal, having presented new methods that can boost the performance of constraint acquisition systems.

I proposed the MQuAcq algorithm which extends QuAcq to acquire all the violated constraints from a negative example, just like MultiAcq does, but with a better complexity bound in terms of the number of queries. I also proposed an optimization on the process of locating scopes that, as experiments demonstrate, helps reduce the number of queries up to 85% in some cases. Another contribution of this work is that I focus on query generation which is a very important but rather overlooked part of the acquisition process. I proposed several heuristics that can be applied during query generation to boost the performance of constraint acquisition algorithms. Also I proposed a technique to exploit the structure of the problem while acquiring constraints.

As future directions, I aim to examine the convergence of acquisition algorithms in the the more general learning model where the queries posted are prone to errors and omissions. This is very important, as it is very likely to happen in real applications. Finally, I intend to integrate the developed methods, as well as other state-of-the-art techniques, in a software tool that will be available for anyone that wishes to model combinatorial problems as CSPs.

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. O'Sullivan, B.: Automated modelling and solving in constraint programming. In: AAAI Conference on Artificial Intelligence. (2010) 1493–1497
5. De Raedt, L., Passerini, A., Teso, S.: Learning constraints from examples. In: Proceedings in Thirty-Second AAAI Conference on Artificial Intelligence. (2018)
6. Bessiere, C., Koriche, F., Lazaar, N., O'Sullivan, B.: Constraint acquisition. Artificial Intelligence **244** (2017) 315–342
7. 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
8. 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
9. 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
10. 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
11. Bshouty, N.: Exact learning boolean functions via the monotone theory. Information and Computation **123**(1) (1995) 146 – 153
12. Bshouty, N.H.: Exact learning from an honest teacher that answers membership queries. Theoretical Computer Science **733** (2018) 4–43
13. Angluin, D.: Queries and concept learning. Machine learning **2**(4) (1988) 319–342
14. Bessiere, C., Coletta, R., O'Sullivan, B., Paulin, M., et al.: Query-driven constraint acquisition. In: International Joint Conference on Artificial Intelligence (IJCAI). Volume 7. (2007) 50–55
15. Shchekotykhin, K., Friedrich, G.: Argumentation based constraint acquisition. In: Ninth IEEE International Conference on Data Mining, IEEE (2009) 476–482
16. 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
17. 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

18. Daoudi, A., Mechqrane, Y., Bessiere, C., Lazaar, N., Bouyakhf, E.H.: Constraint acquisition using recommendation queries. In: International Joint Conference on Artificial Intelligence (IJCAI). (2016) 720–726
19. Bessiere, C., Coletta, R., Daoudi, A., Lazaar, N., Mechqrane, Y., Bouyakhf, E.H.: Boosting constraint acquisition via generalization queries. In: European Conference on Artificial Intelligence (ECAI). (2014) 99–104
20. Arcangioli, R., Bessiere, C., Lazaar, N.: Multiple constraint aquisition. In: International Joint Conference on Artificial Intelligence (IJCAI). (2016) 698–704
21. Paulin, M., Bessiere, C., Sallantin, J.: Automatic design of robot behaviors through constraint network acquisition. In: 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI). Volume 1., IEEE (2008) 275–282
22. 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)
23. Tsouros, D.C., Stergiou, K., Bessiere, C.: Structure-driven multiple constraint acquisition. In: 25th International Conference on Principles and Practice of Constraint Programming. (2019)