Step-wise Explaining How to Solve Constraint Satisfaction Problems

Emilio Gamba[0000−0003−1720−9428], Bart Bogaerts[0000−0003−3460−4251], and Tias Guns[0000−0002−2156−2155]

Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium
{firstname.lastname}@vub.be

Abstract. We investigate the problem of step-wise explaining how to solve constraint satisfaction problems. More specifically, we study how to explain the inference steps that one can take during propagation. The main challenge is finding a sequence of *simple* explanations, where each explanation should aim to be cognitively as easy as possible for a human to verify and understand. This contrasts with the arbitrary combination of facts and constraints that the solver may use when propagating. We identify the explanation-production problem of finding the best sequence of explanations for the maximal consequence of a CSP. We propose the use of a cost function to quantify how simple an individual explanation of an inference step is. Our proposed algorithm iteratively constructs the explanation sequence, agnostic of the underlying constraint propagation mechanisms, by using an optimistic estimate of the cost function, to guide the search for the best explanation at each step. Using reasoning by contradiction, we develop a mechanism to break the most difficult steps up and give the user the ability to *zoom in* on specific parts of the explanation.

Keywords: Explainable Artificial Intelligence · Constraint Solving · Explanation · Automated Reasoning

1 Introduction

In the last few years, as AI systems employ more advanced reasoning mechanisms and computation power, it becomes increasingly difficult to understand why certain decisions are made. Explainable (XAI), a subfield of AI, aims to fulfill the need for trustworthy AI systems to understand *how* and *why* the system made a decision, e.g. for verifying correctness of the system, as well as to control for biased or systematically unfair decisions.

Explanations have been investigated in constraint solving before, most notably for explaining over-constrained, and hence unsatisfiable, problems to a user. The QuickXplain method [7] for example, uses a dichotomic approach that recursively partitions the constraints to find a minimal conflict set. Many other papers consider the same goal and search for explanations of over-constrainedness [9,13].

Despite the fact that we do not (specifically) aim to explain over-constrained problems, our algorithms will also internally make use of methods to extract a minimal set of conflicting constraints often called a *<u>M</u>inimal <u>U</u>nsatisfiable <u>S</u>ubset* (MUS) or *Minimal Unsatisfiable Core* [10].

While explainability of constraint optimisation has received little attention so far, in the related field of *planning*, there is the emerging subfield of *eXplainable AI planning* (XAIP) [5], which is concerned with building planning systems that can explain their own behaviour. This includes answering queries such as "why did the system (not) make a certain decision?", "why is this the best decision?", etc. In contrast to explainable machine learning research [6], in explainable planning one can make use of the explicit *model-based representation* over which the reasoning happens. Likewise, we will make use of the constraint specification available to constraint solvers, more specifically typed first-order logic [12].

This research fits within the general topic of Explainable Agency [8], whereby in order for people to trust autonomous agents, the latter must be able to *explain their decisions* and the *reasoning* that produced their choices. To provide the constraint solver with Explainable Agency [8], we first formalize the problem of step-wise explaining the propagation of a constraint solver through a sequence of small inference steps. Next, we use an optimistic estimate of a given cost function quantifying human interpretability to guide the search to *simple*, low-cost, explanations thereby making use of minimal unsatisfiable subsets. We extend this approach using *reasoning by contradiction* to produce additional explanations of still difficult-to-understand inference steps. Finally, we discuss the challenges and some outlooks to explaining how to solve constraint satisfaction problems.

Publication history This workshop paper is an extended abstract of previous papers presented at workshops and conferences [4,3,1] and a journal paper under review [2].

2 Background and Problem definition

The overarching goal of this paper is to generate a sequence of small reasoning steps, each with an interpretable explanation, and for that we introduce the necessary background.

A *(partial) interpretation* I is defined as a finite set of literals , i.e., expressions of the form $P(\overline{d})$ or $\neg P(\overline{d})$ where P is a relation symbol typed $T_1 \times \cdots \times T_n$ and $\overline{d}$ is a tuple of domain elements where each d_i is of type T_i. For example, $eat(T_{person},\ T_{food})$ defines a relation linking an entity of type person with an entity of type food, if a person eats a certain kind of food. If 'Sam ate pizza, and Luke did not eat rice', then the clue can be interpreted as the partial interpretation $I_{Sam-Luke} = \{eat(Sam,\ Pizza), \neg eat(Luke,\ Rice)\}$.

A partial interpretation is *consistent* if it does not contain both an atom and its negation. It is called a *full* interpretation if it either contains $P(\overline{d})$ or $\neg P(\overline{d})$ for each well-typed atom $P(\overline{d})$.

In the context of first-order logic, the task of finite-domain constraint solving is better known as *model expansion* [11]: given a logical theory T (corresponding to the constraint specification) and a partial interpretation I with a finite domain (corresponding to the initial domain of the variables), find a model M more precise than I (a partial solution that satisfies T).

We define the **maximal consequence** of a theory T and partial interpretation I (denoted $max(I, T)$) as the precision-maximal partial interpretation I_n such that $I \wedge T \models I_n$. More precisely, I_n corresponds to the intersection of all CSP solutions.

2.1 Simple Explanation

Let I_{i-1} and I_i be partial interpretations such that $I_{i-1} \wedge T \models I_i$. We say that (E_i, S_i, N_i) *explains* the derivation of I_i from I_{i-1} if the following holds:

- $N_i = I_i \setminus I_{i-1}$ (i.e., N_i consists of all newly derived facts),
- $E_i \subseteq I_{i-1}$ (i.e., the explaining facts are a subset of what was previously derived),
- $S_i \subseteq T$ (i.e., a subset of the constraints used), and
- $S_i \wedge E_i \models N_i$ (i.e., all newly derived information following from this explanation).

Part of our goal of finding easy to interpret explanations is to avoid redundancy. That is, we want a non-redundant explanation (E_i, S_i, N_i) where none of the facts in E_i or constraints in S_i can be removed while still explaining the derivation of I_i from I_{i-1}; in other words: the explanation must be *subset-minimal*. While subset-minimality ensures that an explanation is non-redundant, it does not quantify how *interpretable* a explanation is. For this, we will assume the existence of a cost function $f(E_i, S_i, N_i)$ that quantifies the interpretability of a single explanation.

Formally, for a given theory T, a cost function f and initial partial interpretation I_0, the **explanation-production problem** consists of finding a non-redundant explanation sequence for (I, T)

$$\langle (I_0, (\emptyset, \emptyset, \emptyset)), (I_1, (E_1, S_1, N_1)), \ldots, (I_n, (E_n, S_n, N_n)) \rangle$$

such that a predefined aggregate[1] over the sequence $(f(E_i, S_i, N_i))_{i \leq n}$ is minimised.

Consider the following problem of 3 persons going to a restaurant ordering food and drinks, but we do not know the orders:

1. "Sam decides to eat pizza."
2. "The one who ate rice drank Tea."
3. "Pasta does not go well with Juice."
4. "John orders water, and Luke always drinks Tea."

We extend the problem statement with the following relations:

$$\{eat(T_{person},\ T_{food}),\ drink(T_{person},\ T_{drinks}),\ match(T_{food},\ T_{drinks})\}$$

[1] An aggregate like $max()$ will moderate the most difficult step, while $average()$ enforces an overall simpler explanation sequence.

Every type has its corresponding entities:

- $T_{food} = \{Pizza, Rice, Pasta\}$;
- $T_{person} = \{Sam, Luke, John\}$
- $T_{drinks} = \{Tea, Juice, Water\}$

Furthermore, we use a cost function f which *favours the use of simple constraints* (ex: interpretation of a clue, simple inference techniques such as bijectivity[2] or transitivity[3]) and *penalizes the use of combination of constraints* (combining a clue with an inference mechanism).

In the beginning, the partial interpretation I_0 is empty. From the clue (1), we can derive a new fact relating Sam and pizza $I_1 = \{eat(Sam, Pizza)\}$. In fact, we can also derive that Sam did not eat *Pasta* and *Rice*. Formally, the tuple (E_1, S_1, N_1) explains the inference step I_1 to I_2, where

- $E_1 = \{\ eat(Sam, Pizza)\ \}$
- $S_1 = \{\ \forall\ p\ \in\ T_{person},\ \exists!\ f\ \in\ T_{food} :\ eat(p,\ f)\ \}$ (bijectivity)
- $N_1 = \{\ \neg eat(Sam, Pasta), \neg eat(Sam, Rice)\ \}$

2.2 Nested Explanation

Each explanation in the sequence will be non-redundant and hence as small as possible. Yet, in our earlier work some explanations were still quite hard to understand, mainly since multiple constraints had to be combined with a number of previously derived facts. We propose the use of simple *nested* explanations using reasoning by contradiction, hence reusing the techniques from previous section.

Given a non-trivial explanation (E, S, N), a nested explanation starts from the explaining facts E, and the counterfactual assumption of the negation of a newly derived fact. At each step, it only uses clues from S and each step is easier to understand (has a strictly lower cost) than the parent explanation which has cost $f(E, S, N)$. A contradiction is then derived from the counterfactual assumption. Each of the reasoning steps leading to the contradiction are what constitutes the nested explanation sequence.

3 Explanation-Producing search

Ideally, we could generate all explanations of each fact in $max(I_0, T)$, and search for the lowest scoring sequence among those explanations. However, the number of explanations for each fact quickly explodes with the number of constraints, and is hence not feasible to compute. Instead, we will iteratively construct the sequence, by generating candidates for a given partial interpretation and searching for the 'easiest' one among those.

[2] Each entity of one type is linked to exactly one entity of each other type.

[3] The entities are logically linked, for example: If $eat(Sam,\ Pizza)$ and $match(Pizza,\ Juice)$, then consistently Sam should be consistently linked with Juice $drink(Sam,\ Juice)$

The task of finding a non-redundant explanation itself can be reduced to finding a Minimal Unsat Subset (MUS), which means that, whenever one of the facts in E or constraints in S is removed, the result is no longer an explanation. More formally, E and S form a minimal set that entail n if and only if $\{E \wedge S \wedge \neg n\}$ is a minimal unsatisfiable set.

Using an optimistic estimate of the cost function f, we guide the search towards the next *cost-minimal non-redundant* explanation $(E \subseteq I, S \subseteq T, \{n\})$ that explains n (and possibly explains more).

We refer to [1] for further details on the explanation-producing algorithm and [2] introducing the concept of what we call nested explanation sequences.

4 Discussion and Future work

In terms of *efficiency*, the main bottleneck of the current algorithm is the search towards the next cost-minimal explanation. More precisely, generating candidate explanations requires repeatedly searching for a MUS for increasing constraint sets, which is a hard problem by itself. Therefore, in future work we want to investigate unsat-core *optimization* with respect to a cost-function, as well as exploring other heuristics to construct non-redundant explanation sequences.

References

1. Bogaerts, B., Gamba, E., Claes, J., Guns, T.: Step-wise explanations of constraint satisfaction problems. In: 24th European Conference on Artificial Intelligence (ECAI) (2020), accepted
2. Bogaerts, B., Gamba, E., Guns, T.: A framework for step-wise explaining how to solve constraint satisfaction problems (2020)
3. Claes, J., Bogaerts, B., Canoy, R., Gamba, E., Guns, T.: Zebratutor: Explaining how to solve logic grid puzzles. In: Proceedings of BNAIC and Benelearn. CEUR Workshop Proceedings, vol. 2491 (2019)
4. Claes, J., Bogaerts, B., Canoy, R., Guns, T.: User-oriented solving and explaining of natural language logic grid puzzles. In: The Third Workshop on Progress Towards the Holy Grail (2019)
5. Fox, M., Long, D., Magazzeni, D.: Explainable planning. In: IJCAI'17 workshop on Explainable AI (arXiv:1709.10256)
6. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) **51**(5), 1–42 (2018)
7. Junker, U.: Quickxplain: Conflict detection for arbitrary constraint propagation algorithms. In: IJCAI'01 Workshop on Modelling and Solving problems with constraints (2001)
8. Langley, P., Meadows, B., Sridharan, M., Choi, D.: Explainable agency for intelligent autonomous systems. In: Twenty-Ninth IAAI Conference (2017)
9. Leo, K., Tack, G.: Debugging unsatisfiable constraint models. In: International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. pp. 77–93. Springer (2017)

10. Marques-Silva, J.: Minimal unsatisfiability: Models, algorithms and applications. In: 2010 40th IEEE International Symposium on Multiple-Valued Logic. pp. 9–14. IEEE (2010)
11. Mitchell, D.G., Ternovska, E., Hach, F., Mohebali, R.: Model expansion as a framework for modelling and solving search problems. Tech. Rep. TR 2006-24, Simon Fraser University, Canada (2006)
12. Wittocx, J., Denecker, M., Bruynooghe, M.: Constraint propagation for first-order logic and inductive definitions. ACM Trans. Comput. Log. **14** (2013)
13. Zeighami, K., Leo, K., Tack, G., de la Banda, M.G.: Towards semi-automatic learning-based model transformation. In: Proceedings of CP. pp. 403–419 (2018)