Discrete Applied Mathematics 184 (2015) 89–113

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Characterising the complexity of constraint satisfaction problems defined by 2-constraint forbidden patterns☆

Martin C. Cooper[a], Guillaume Escamocher[b,*]

[a] IRIT, University of Toulouse, France

[b] INSIGHT, University College Cork, Ireland

ARTICLE INFO

Article history:
Received 7 October 2013
Received in revised form 24 September 2014
Accepted 18 October 2014
Available online 20 November 2014

Keywords:
Constraint satisfaction problem
Tractable class
Forbidden pattern

ABSTRACT

Although the CSP (constraint satisfaction problem) is NP-complete, even in the case when all constraints are binary, certain classes of instances are tractable. We study classes of binary CSP instances defined by excluding subproblems. This approach has recently led to the discovery of novel tractable classes. The complete characterisation of all tractable classes defined by forbidding patterns (where a pattern is simply a compact representation of a set of subproblems) is a challenging problem. We demonstrate a dichotomy in the case of forbidden patterns consisting of either one or two constraints. This has allowed us to discover several new tractable classes including, for example, a novel generalisation of 2SAT. We then extend this dichotomy to existential patterns which are only forbidden on specific domain values.

1. Introduction

In this paper we study the generic combinatorial problem known as the binary constraint satisfaction problem (CSP) in which the aim is to determine the existence of an assignment of values to n variables such that a set of constraints on pairs of variables are simultaneously satisfied. The generic nature of the CSP has led to diverse applications, notably in the fields of Artificial Intelligence and Operations Research [24].

A fundamental research question in complexity theory is the identification of tractable subproblems of NP-complete problems. Classical approaches have consisted in identifying types of constraints which imply the existence of a polynomial-time algorithm. Among the most well-known examples, we can cite linear constraints and Horn clauses. In an orthogonal approach, restrictions are placed solely on the (hyper)graph of constraint scopes, known as the constraint (hyper)graph. In some cases, dichotomies have even been proved characterising all tractable classes definable by placing restrictions either on the constraint relations [4,3,2] or on the constraint (hyper)graph [21–23].

Recently, a new avenue of research has been investigated: the identification of tractable classes of CSP instances defined by forbidding a specific (set of) subproblems. Novel tractable classes have been discovered by forbidding simple 3-variable subproblems [13,15]. A dichotomy has even been discovered for classes of binary CSP instances defined by forbidding configurations of incompatibilities [5].

One concrete example of a tractable class defined by forbidding a generic subproblem (known as a pattern) is the set of binary CSP instances satisfying the broken-triangle property [13]: a binary CSP instance on variables $X_1, \ldots, X_n$ satisfies the

☆ Supported by ANR Project ANR-10-BLAN-0210.

* Corresponding author.
E-mail addresses: cooper@irit.fr (M.C. Cooper), guillaume.escamocher@insight-centre.org (G. Escamocher).

http://dx.doi.org/10.1016/j.dam.2014.10.035
0166-218X/

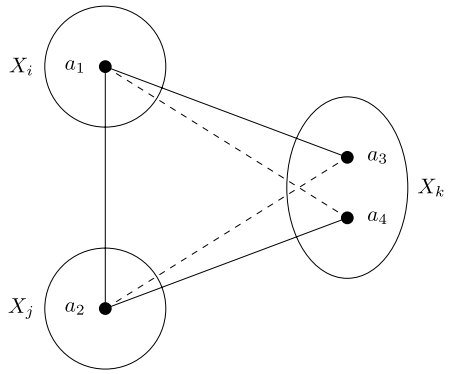

Fig. 1. Pattern forbidden by the broken-triangle property.

broken-triangle property if $\forall i < j < k \in \{1, \ldots, n\}$, whenever the assignments $a_1 = \langle X_i, a\rangle$, $a_2 = \langle X_j, b\rangle$, $a_3 = \langle X_k, c\rangle$, $a_4 = \langle X_k, d\rangle$ are such that the pairs of assignments (a_1, a_2), (a_1, a_3), (a_2, a_4) are compatible, then at least one of the pairs of assignments (a_1, a_4), (a_2, a_3) is also compatible. The forbidden subproblem is shown in Fig. 1. It has three constraints, because it has at least one edge between each of the three different pairs of variables. For example, any binary CSP instance whose constraint graph is a tree satisfies the broken-triangle property for some ordering of its variables; furthermore such an ordering can be determined in polynomial time. However, tractability is not due to a property of the constraint graph, since instances satisfying the broken-triangle property exist for arbitrary constraint graphs.

Recently the broken-triangle property has also inspired the development of simplification operations based on the absence of patterns of compatibilities and incompatibilities on particular variables or values (known as existential patterns). While in the present paper we infer tractability from a globally-held property (that is, a given pattern does not appear anywhere in the instance), [6] show that even with only local properties of the same kind (a given pattern does not appear on a given variable X_i) it is possible to deduce information about the relationship between the variable X_i and the rest of the CSP instance. Depending on that information, it may then be possible to remove the variable without modifying the satisfiability of the CSP instance. Note that any pattern permitting this kind of elimination also defines a tractable class when it does not occur on any variable, but not all tractable patterns permit variable elimination. It was possible to characterise all patterns permitting variable elimination [6], but for the more challenging problem of characterising all forbidden patterns defining tractable classes, we restrict ourselves in the present paper to 2-constraint patterns as an important first step towards a complete characterisation.

Two other examples of forbidden patterns which define tractable classes of binary CSP instances are based on the transitivity of compatibilities or incompatibilities [16]. The former class consists of all binary CSP instances in which for all triples of assignments $a_1 = \langle X_i, a\rangle$, $a_2 = \langle X_j, b\rangle$, $a_3 = \langle X_k, c\rangle$ to three distinct variables, whenever the pairs (a_1, a_2), (a_2, a_3) are both compatible, the third pair (a_1, a_3) is also compatible. The latter class consists of all binary CSP instances in which for all triples of assignments $a_1 = \langle X_i, a\rangle$, $a_2 = \langle X_j, b\rangle$, $a_3 = \langle X_k, c\rangle$ to three distinct variables, whenever the pairs (a_1, a_2), (a_2, a_3) are both incompatible, the third pair (a_1, a_3) is also incompatible. This property is satisfied, for example, by instances consisting of unary constraints and non-overlapping AllDifferent constraints (since $a = b \wedge b = c \Rightarrow a = c$). The class of binary CSP instances satisfying this negative-transitivity property has been generalised to a large tractable class of optimisation problems involving cost functions of arbitrary arity [15,16].

Any class of instances defined by a forbidden pattern is necessarily recognisable in polynomial time by a simple exhaustive search for the pattern.

The present paper provides an essential first step towards the identification of all tractable classes defined by forbidding patterns, namely a dichotomy for the special case of 2-constraint forbidden patterns. This investigation of small forbidden patterns has already allowed us to uncover several novel tractable classes. We expect that this dichotomy will in the future represent an important base case in a more general characterisation of tractable classes of constraint problems defined by local structure. The tractable classes described in this paper may prove to be the inspiration for larger tractable classes of general-arity CSPs or may lead to the development of simplification rules for CSPs.

The paper is structured as follows. In Sections 2–4 we give the necessary definitions concerning patterns, tractability and reductions between patterns, together with some preprocessing operations which can always be assumed to have been applied. In Section 5 we give a dichotomy for one-constraint patterns. Section 6 is devoted to the main dichotomy result for two-constraint patterns. This result first appeared as a conference paper without the proof of the most difficult case [12]. Finally, we extend this dichotomy to include existential patterns in Section 7.

2. Definitions

We first define the notion of a CSP pattern. A pattern can be seen as a generalisation of a binary CSP instance; it represents a *set* of subproblems by leaving the consistency of some tuples undefined. We use the term *point* to denote an assignment