Artificial Intelligence 234 (2016) 196–218

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Broken triangles: From value merging to a tractable class of general-arity constraint satisfaction problems

Martin C. Cooper [a,*], Aymeric Duchein [a], Achref El Mouelhi [b], Guillaume Escamocher [c], Cyril Terrioux [b], Bruno Zanuttini [d]

[a] *IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, France*
[b] *Aix-Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, Marseille, France*
[c] *INSIGHT Centre for Data Analytics, University College Cork, Cork, Ireland*
[d] *GREYC, UMR 6072, Normandie Université, UNICAEN, CNRS, ENSICAEN, Caen, France*

ARTICLE INFO

Article history:
Received 17 September 2015
Received in revised form 1 February 2016
Accepted 3 February 2016
Available online 4 February 2016

Keywords:
CSP
Constraint satisfaction
Domain reduction
Tractable class
Hybrid tractability
NP-completeness
Global constraints

ABSTRACT

A binary CSP instance satisfying the broken-triangle property (BTP) can be solved in polynomial time. Unfortunately, in practice, few instances satisfy the BTP. We show that a local version of the BTP allows the merging of domain values in arbitrary instances of binary CSP, thus providing a novel polynomial-time reduction operation. Extensive experimental trials on benchmark instances demonstrate a significant decrease in instance size for certain classes of problems. We show that BTP-merging can be generalised to instances with constraints of arbitrary arity and we investigate the theoretical relationship with resolution in SAT. A directional version of general-arity BTP-merging then allows us to extend the BTP tractable class previously defined only for binary CSP. We investigate the complexity of several related problems including the recognition problem for the general-arity BTP class when the variable order is unknown, finding an optimal order in which to apply BTP merges and detecting BTP-merges in the presence of global constraints such as AllDifferent.

1. Introduction

At first sight one could assume that the discipline of constraint programming has come of age. On the one hand, efficient solvers are regularly used to solve real-world problems in diverse application domains while, on the other hand, a rich theory has been developed concerning, among other things, global constraints, tractable classes, reduction operations and symmetry. However, there often remains a large gap between theory and practice, which is perhaps most evident when we look at the large number of deep results concerning tractable classes which have yet to find any practical application. The research reported in this paper is part of a long-term project to bridge the gap between theory and practice. Our aim is not only to develop new tools but also to explain why present tools work so well.

Most research on tractable classes has been based on classes defined by placing restrictions either on the types of constraints [1,2] or on the constraint hyper-graph whose vertices are the variables and whose hyper-edges are the constraint scopes [3,4]. Another way of defining classes of binary CSP instances consists of imposing conditions on the microstruc-

* Corresponding author.
E-mail addresses: cooper@irit.fr (M.C. Cooper), Aymeric.Duchein@irit.fr (A. Duchein), achref.elmouelhi@lsis.org (A. El Mouelhi), guillaume.escamocher@insight-centre.org (G. Escamocher), cyril.terrioux@lsis.org (C. Terrioux), bruno.zanuttini@unicaen.fr (B. Zanuttini).

http://dx.doi.org/10.1016/j.artint.2016.02.001
0004-3702/

ture, a graph whose vertices are the possible variable-value assignments with an edge linking each pair of compatible assignments [5,6]. If each vertex of the microstructure, corresponding to a variable-value assignment $\langle x, a\rangle$, is labelled (or coloured) by the variable x, then this so-called coloured microstructure retains all information from the original instance. The broken-triangle property (BTP) is a simple local condition on the coloured microstructure which defines a tractable class of binary CSP [7]. The BTP corresponds to forbidding a simple pattern, known as a broken triangle, in the coloured microstructure for a given variable order. Inspired by the BTP, investigation of other forbidden patterns in the coloured microstructure has led to the discovery of new tractable classes [8–10] as well as new reduction operations based on variable or value elimination [11,12]. The BTP itself has also been directly generalised in several different ways. For example, it has been shown that under an assumption of strong path consistency, the BTP can be considerably relaxed since not all broken triangles need be forbidden to define a tractable class [13–15]. Indeed, even without any assumptions of consistency, it is not necessary to forbid all broken triangles [12]. Imposing the BTP in the dual problem leads directly to a tractable class of general-arity CSPs [16]. The BTP has also been generalised to the Broken Angle Property which defines a tractable class of Quantified Constraint Satisfaction Problems [17].

In this paper we show that the absence of broken triangles on a pair of values in a domain allows us to merge these two values while preserving the satisfiability of the instance. Furthermore, given a solution to the reduced instance, it is possible to find a solution to the original instance in linear time (Section 3). We then investigate the interactions between arc consistency and BTP-merging operations (Section 4) and show that it is NP-hard to find the best sequence of BTP-merging (and arc consistency) operations (Section 5). The effectiveness of BTP-merging in reducing domains in binary CSP benchmark problems is investigated in Section 6. In the second half of the paper we consider general-arity CSPs. Section 7 shows how to generalise BTP-merging to instances containing constraints of any arity (where all constraints are given in the form of either tables, lists of compatible tuples or lists of incompatible tuples). We then go on to consider global constraints, and in particular the AllDifferent constraint, in Section 8. Finally, a directional version of the general-arity BTP allows us to define a tractable class of general-arity CSP instances which is incomparable with the tractable class obtained by directly imposing the BTP in the dual [16] (Section 9). However, on the negative side, we then show that it is NP-complete to determine the existence of a variable order for which an instance falls into this tractable class. The results of Sections 3, 7, 9 and Sections 4, 5 first appeared in two conference papers (respectively [18] and [19]).

2. The Constraint Satisfaction Problem

For simplicity of presentation we use two different representations of constraint satisfaction problems. In the binary case, our notation is fairly standard, whereas in the general-arity case we use a notation close to the representation of SAT instances. This is for presentation only, though, and our algorithms do *not* need instances to be represented in this manner.

Definition 1. A *binary CSP instance* I consists of

- a set X of n variables,
- a domain $\mathcal{D}(x)$ of possible values for each variable $x \in X$,
- a relation $R_{xy} \subseteq \mathcal{D}(x) \times \mathcal{D}(y)$, for each pair of distinct variables $x, y \in X$, which consists of the set of compatible pairs of values (a, b) for variables (x, y).

A *partial solution* to I on $Y = \{y_1, \ldots, y_r\} \subseteq X$ is a set $\{\langle y_1, a_1\rangle, \ldots, \langle y_r, a_r\rangle\}$ such that $\forall i, j \in [1, r]$, $(a_i, a_j) \in R_{y_i y_j}$. A *solution* to I is a partial solution on X.

For simplicity of presentation, Definition 1 assumes that there is exactly one constraint relation for each pair of variables. The number of constraints e is the number of pairs of variables x, y such that $R_{xy} \neq \mathcal{D}(x) \times \mathcal{D}(y)$. An instance I is *arc consistent* if for each pair of distinct variables $x, y \in X$, each value $a \in \mathcal{D}(x)$ has an AC-support at y, i.e. a value $b \in \mathcal{D}(y)$ such that $(a, b) \in R_{xy}$.

In our representation of general-arity CSP instances, we require the notion of *tuple* which is simply a set of variable-value assignments. For example, in the binary case, the tuple $\{\langle x, a\rangle, \langle y, b\rangle\}$ is *compatible* if $(a, b) \in R_{xy}$ and *incompatible* otherwise.

Definition 2. A *(general-arity) CSP instance* I consists of

- a set X of n variables,
- a domain $\mathcal{D}(x)$ of possible values for each variable $x \in X$,
- a set NoGoods(I) consisting of incompatible tuples.

A *partial solution* to I on $Y = \{y_1, \ldots, y_r\} \subseteq X$ is a tuple $t = \{\langle y_1, a_1\rangle, \ldots, \langle y_r, a_r\rangle\}$ such that no subset of t belongs to NoGoods(I). A *solution* is a partial solution on X.