Constraint Decomposition Based on Tractable Properties

Achref El Mouelhi
Aix-Marseille Université, CNRS, ENSAM, Université de Toulon,
LSIS UMR 7296, 13397 Marseille Cedex 20 - France
achref.elmouelhi@lsis.org

***Abstract*—Identifying tractable classes of constraint satisfaction problems (CSPs) has been studied over the past two decades, and is now a very active research domain. Recently, some works have shown the interest of these tractable classes from a different viewpoint. For example, it is known that if there is no broken triangle on each two values of a given variable in an arc-consistent binary CSP, then this variable can be eliminated without changing its satisfiability. Next, it has been proved that even when this rule cannot be applied, it may be the case that for a given pair of values no broken triangle occurs, if this is the case, then we can perform a domain-reduction operation which consists in merging these values while preserving satisfiability.**

In this paper, we show that partial or total presence of tractable properties in non-binary constraints is sufficient for them to be decomposable into a set of lower arity constraints while preserving equivalence. More precisely, we prove that the presence of bijection or Dual Broken Triangle Property in non-binary constraint permits to decompose it while preserving equivalence.

I. Introduction

Much research has been devoted to identifying properties, called *tractable properties* under which we can solve CSP in polytime (the class of CSPs satisfying one of these properties is called *tractable class*). Most of this area of research includes three main directions. The first direction studies the structural properties of the constraint network. For example, it is well-known that tree-structured binary CSP can be solved in linear time no matter the associated relations [1]. The second direction concerns the language defining the constraints, i.e. the value combinations related to constraints. For example, bijection constraints [2] can also be solved in linear time [3]. The last direction investigates both the two last kind properties, such as *Broken Triangle Property* (BTP) [4] and its extensions DBTP [5], k-BTP [6], WBTP [7] or m-wBTP [8]. Except BTP and its extensions, most tractable classes rarely occur in practice, which diminish their interest (see [9]). So, some other work has focused on studying another way to exploit tractable properties. In [3], the authors introduced the notion of variable elimination based on tractable properties. Indeed, they proved that partial presence of *functional* configuration permits *variable elimination* to give a new CSP which leaves the solution of the original CSP unchanged. In addition, Cooper et al. in [4] proved that if there is no broken triangle on each two values of a given variable in an arc-consistent binary CSP, then this variable can be eliminated while preserving satisfiability. More recently, it has been proved that the absence of broken triangles on two values for a given variable in a binary CSP allows us to merge them while preserving satisfiability.

In this work we continue this line of research and we establish some links between tractable properties and constraint decomposition, i.e. we apply some properties related to tractable classes for checking the decomposability of constraints. More precisely, we show that whether two or more variables in the scope of a given non-binary constraint satisfy bijection, then so makes it decomposable into two constraints of arity less than that of original one. We also prove that the absence of dual broken triangle w.r.t. any partition of variable in the scope of a given constraint allows us to decompose it into three lower arity constraints. In fact, decreasing the arity of constraint could have a substantial effect on the processing complexity. For example, given a binary CSP, checking whether a variable ordering exists for BTP can be achieved in polytime [4]. However, given a non-binary CSP, checking whether a variable ordering for DGABTP (the generalization of BTP to CSPs with arbitrary arity) exists is NP-complete [10]. Finally constraint decomposition may be considered as a transformation leading to some tractable classes. Indeed, in [9], the authors introduce the concept of transformation (e.g. filtering by consistency, decomposition, binary encoding...) of CSPs and show its interest especially when translated CSP belong to a known tractable class.

The remainder of this paper is organized as follows. In Section 2, we give formal background on constraint satisfaction problems. Then we examine the cases of constraint decomposition using bijection. In section 4, we show that the absence of dual broken triangle in a given constraint prove that it is decomposable. Finally, we give a discussion and perspective for future work.

II. Formal background

We first define constraint satisfaction problems (CSPs).

Definition 1 (CSP): A constraint satisfaction problem is a triple $P = (X, D, C)$, with

- $X = \{x_1, ..., x_n\}$: a set of n **variables**,
- $D = \{D(x_1), ..., D(x_n)\}$: a set of finite **domains**,
- $C = \{c_1, ..., c_e\}$: a set of e **constraints**. Each constraint c_i is a couple $(S(c_i), R(c_i))$ where:
  - $S(c_i) = \{x_{i_1}, ..., x_{i_{a_i}}\} \subseteq X$, is the **scope** of c_i,
  - $R(c_i) \subseteq D(x_{i_1}) \times ... \times D(x_{i_{a_i}})$, is the associated **relation** which allows r tuples.

We assume that each variable is at least in the scope of one constraint and if there are two constraints with the same scope then they will be merged. $|S(c_i)|$ is the *arity* of the constraint c_i (i.e. the number of variables in the constraint scope) and it will be denoted a_i. A CSP is said to be *binary* if the arity of each constraint is two (in this case, we denote $c_{i,j}$ the constraint with $S(c_{i,j}) = \{x_i, x_j\}$), otherwise it is *non-binary* (*n-ary* or *of arbitrary arity*). A CSP is *ternary* when the arity of each constraint is less than or equal to three.

We continue with the following notations which are necessary for the rest.

Notation 1 (tuple and relation projection): Given a constraint c_i, a tuple $t \in R(c_i)$ and a set of variables $\{x_{i_1}, ..., x_{i_k}\} \subseteq S(c_i)$: $t[\{x_{i_1}, ..., x_{i_k}\}] = (v_j \in t \mid x_j \in \{x_{i_1}, ..., x_{i_k}\})$ is the **projection** of the tuple t onto $\{x_{i_1}, ..., x_{i_k}\}$. $R(c_i)[\{x_{i_1}, ..., x_{i_k}\}] = \{t[\{x_{i_1}, ..., x_{i_k}\}] \mid t \in R(c_i)\}$ is the projection of $R(c_i)$ on $\{x_{i_1}, ..., x_{i_k}\}$.

Notation 2 (constraint restriction): Given a constraint c_i, $c_i[\{x_{i_1}, ..., x_{i_k}\}]$ is the restriction of c_i on $\{x_{i_1}, ..., x_{i_k}\}$ ($\subseteq S(c_i)$). If we denote $c_\ell = c_i[\{x_{i_1}, ..., x_{i_k}\}]$, then $S(c_\ell) = \{x_{i_1}, ..., x_{i_k}\}$ and $R(c_\ell) = R(c_i)[\{x_{i_1}, ..., x_{i_k}\}]$.

Given a CSP P, solving P consists of finding a solution, i.e. assigning one value per domain for each variable without violating any constraint. This problem is known to be NP-hard for both binary and non-binary case. We now recall the definition of some tractable classes which will be necessary for the rest. The first concerns the bijection.

Definition 2 (bijection [2]): A binary relation $R(c_{i,j})$ is called **bijection** if each value in $D(x_i)$ is only compatible with one value in $D(x_j)$ (and conversely). We can also say the constraint c_ℓ satisfies bijection.

We now turn to the definition of Dual Broken Triangle Property covering CSPs of arbitrary arity.

Definition 3 (DBTP [5]): A CSP $P = (X, D, C)$ satisfies **DBTP** (for Dual broken triangle Property) w.r.t. the constraint ordering $\prec$ iff for all triples of constraints (c_i, c_j, c_k) s.t. $c_i \prec c_j \prec c_k$, for all $t_i \in R(c_i)$, $t_j \in R(c_j)$ and $t_k, t'_k \in R(c_k)$ s.t.

- $t_i[S(c_i) \cap S(c_j)] = t_j[S(c_i) \cap S(c_j)]$
- $t_i[S(c_i) \cap S(c_k)] = t_k[S(c_i) \cap S(c_k)]$
- $t'_k[S(c_j) \cap S(c_k)] = t_j[S(c_j) \cap S(c_k)]$

then

- either $t'_k[S(c_i) \cap S(c_k)] = t_i[S(c_i) \cap S(c_k)]$
- or $t_j[S(c_j) \cap S(c_k)] = t_k[S(c_j) \cap S(c_k)]$.

When $t'_k[S(c_i) \cap S(c_k)] \neq t_i[S(c_i) \cap S(c_k)]$ and $t_j[S(c_j) \cap S(c_k)] \neq t_k[S(c_j) \cap S(c_k)]$, we say that we have a *Dual broken triangle* (DBT) on c_k and it will be denoted (t'_k, t_j, t_i, t_k).

We now conclude this part with the following notation:

Notation 3: x_K denotes a non-empty subset of a_K variables of X ($\{\{x_{k_1}, ..., x_{k_{a_K}}\} \mid \forall 1 \leq i \leq a_K, x_{k_i} \in X\}$) and v_K denotes a tuple $(v_{k_1}, ..., v_{k_{a_K}})$ containing one value for each variable in x_K.

III. Decomposition based on bijection

In this section, we prove that the presence of bijection in non-binary constraint is a sufficient condition to be decomposable.

Theorem 1: Given a non-binary constraint c_ℓ, if there are $x_i, x_j \in S(c_\ell)$ s.t. $R(c_\ell)[\{x_i, x_j\}]$ is a bijection, then c_ℓ can be decomposed into two constraints without loss of equivalence:

- either $c_{i,j}$ and c'_ℓ with $S(c'_\ell) = S(c_\ell) \setminus \{x_i\}$
- or $c_{i,j}$ and c''_ℓ with $S(c''_\ell) = S(c_\ell) \setminus \{x_j\}$

Proof: Let c_ℓ be a non-binary constraint and $x_i, x_j \in S(c_\ell)$ s.t. $R(c_\ell)[\{x_i, x_j\}]$ is a bijection. We will prove by contradiction that replacing c_ℓ by $c_{i,j}$ and c'_ℓ with $S(c'_\ell) = S(c_\ell) \setminus \{x_i\}$ (similarly if we replace c_ℓ by $c_{i,j}$ and c''_ℓ with $S(c''_\ell) = S(c_\ell) \setminus \{x_j\}$) preserves equivalence. For this, we suppose that there is an assignment $\mathcal{A} = (..., v_i, ..., v_j, ...)$ which does not violate any decomposed constraint (obtained after decomposition) but does not satisfy the original constraint c_ℓ. We can express it as follows:

- $\mathcal{A}[\{x_i, x_j\}] \in R(c_{i,j})$ and
- $\mathcal{A}[S(c'_\ell)] \in R(c'_\ell)$ but
- $\mathcal{A} \notin R(c_\ell)$

So there are two tuples $t_\ell, t'_\ell \in R(c_\ell)$ s.t.:

- $t_\ell[\{x_i, x_j\}] = \mathcal{A}[\{x_i, x_j\}]$ and $t_\ell[S(c'_\ell)] \neq \mathcal{A}[S(c'_\ell)]$
- $t'_\ell[\{x_i, x_j\}] \neq \mathcal{A}[\{x_i, x_j\}]$ and $t'_\ell[S(c'_\ell)] = \mathcal{A}[S(c'_\ell)]$

$t'_\ell[\{x_j\}]$ must be different to $t_\ell[\{x_j\}]$, otherwise $\mathcal{A} \in R(c_\ell)$. In this case, $\mathcal{A}[\{x_i\}]$ is compatible with both $t_\ell[\{x_j\}]$ ($= \mathcal{A}[\{x_j\}]$) and $t'_\ell[\{x_j\}]$ ($\neq \mathcal{A}[\{x_i\}]$). Since $R(c_\ell)[\{x_i, x_j\}]$ is a bijection, so it is impossible that a value $\in D(x_i)$ is compatible with two different values $\in D(x_j)$. □

Example 1: The first table shows a decomposable constraint c_ℓ because $R(c_\ell)[\{x_i, x_j\}]$ is a bijection. The second illustrates two constraints obtained after decomposing the first one.

$R(c_\ell)$			
x_i	x_j	x_m	x_k
a	b	c	d
a'	b'	c'	d
a''	b''	c''	d'

$R(c_{i,j})$	$R(c'_\ell)$
$x_i\ x_j$	$x_i\ x_m\ x_k$
$a\ b$	$a\ c\ d$
$a'\ b'$	$a'\ c'\ d$
$a''\ b''$	$a''\ c''\ d'$

A. Generalizing bijection to CSPs of arbitrary arity

We now generalize bijection to CSPs of arbitrary arity. Before that, we must point out that a first attempt was made in [11] and it represents a simple extension but not a generalization.

Definition 4 (G-bijection): Given a non-binary constraint c_ℓ, the relation $R(c_\ell)$ satisfies **G-bijection** (for Generalalized bijection) if there are two disjoint subsets x_I and x_J with $S(c_\ell) \supseteq x_I \cup x_J$ s.t. $\forall v_I \in R(c_\ell)[V_I]$, there is a unique $v_J \in R(c_\ell)[x_J]$ such as $(v_I, v_J) \in R(c_\ell)$. We can also say the constraint c_ℓ satisfies G-bijection.

If $S(c_\ell) = x_I \cup x_J$ we can speak of total presence of G-bijection in constraint c_ℓ, otherwise we speak of partial presence. We now give the link between partial presence of G-bijection and decomposability.

Theorem 2: Given a constraint c_ℓ of arbitrary arity, if there are three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$ s.t. $c_\ell[x_I \cup x_J]$ satisfies G-bijection, then c_ℓ can be decomposed into two constraints c'_ℓ and c''_ℓ (with $S(c'_\ell) = x_I \cup x_J$ and $S(c''_\ell) = x_I \cup x_K$) without losing equivalence.

Proof: Let C_ℓ be a constraint of arbitrary arity and three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$ s.t. $c_\ell)[x_I \cup x_J]$ satisfies G-bijection. We will prove by contradiction that replacing c_ℓ by c'_ℓ and c''_ℓ (with $S(c'_\ell) = x_I \cup x_J$ and $S(c''_\ell) = x_I \cup x_K$) preserves equivalence. For this, we suppose that there is an assignment $\mathcal{A} = (v_I, v_J, v_K)$ which does not violate c'_ℓ neither c''_ℓ but does not satisfy the original constraint c_ℓ. We can express it as follows:

- $\mathcal{A}[x_I \cup x_J] \in R(c'_\ell)$ and
- $\mathcal{A}[x_I \cup x_K] \in R(c''_\ell)$ but
- $\mathcal{A} \notin R(c_\ell)$

So there are two tuples $t_\ell, t'_\ell \in R(c_\ell)$ s.t.:

- $t_\ell[x_I \cup x_J] = \mathcal{A}[x_I \cup x_J] = (v_I, v_J)$ and $t_\ell[x_I \cup x_K] = \mathcal{A}[x_I \cup x_K] \neq (v_I, v_K)$
- $t'_\ell[x_I \cup x_K] = \mathcal{A}[x_I \cup x_K] = (v_I, v_K)$ and $t'_\ell[x_I \cup x_J] = \mathcal{A}[x_I \cup x_J] \neq (v_I, v_J)$

$t'_\ell[x_J]$ must be different to $t_\ell[x_J]$, otherwise $\mathcal{A} \in R(c_\ell)$. Similarly for $t'_\ell[x_K]$ which should be different to $t_\ell[x_K]$. In this case, $\mathcal{A}[x_I]$ is compatible with both $t_\ell[x_J]$ ($= \mathcal{A}[x_J]$) and $t'_\ell[x_J]$ ($\neq \mathcal{A}[x_I]$). Since $R(c_\ell)[x_I \cup x_J]$ is a G-bijection, so it is impossible that v_I (which is $\mathcal{A}[x_I]$) is compatible with $t'_\ell[x_J]$ and $t_\ell[x_J]$ at the same time. □

Example 2: Given a constraint c_ℓ and the partition $x_I = \{x_i, x_m\}$, $x_J = \{x_j, x_s\}$ and $x_K = \{x_k\}$, the constraint $c_\ell[x_I \cup x_J]$ satisfies G-bijection and also it is decomposable into two constraints c'_ℓ and c''_ℓ with $S(c'_\ell) = \{x_i, x_m, x_j, x_s\}$ and $S(c''_\ell) = \{x_i, x_m, x_k\}$.

$R(c_\ell)$				
x_i	x_m	x_j	x_s	x_k
a	b	c'	d'	e
a	b'	c	d'	e'
a'	b	c'	d	e'
a'	b'	c	d	e'
a''	b	c''	d	e'

If we consider the partition $x_I = \{x_i, x_m\}$ and $x_J = \{x_j, x_s, x_k\}$, c_ℓ satisfies G-bijection.

Based on Example 2, one can suppose that if c_ℓ satisfies G-bijection w.r.t. a given partition of variables in its scope x_I and x_J then $c_\ell[x_I \cup x'_J]$ and $c_\ell[x'_I \cup x_J]$ also satisfy G-bijection (with $x'_I \subset x_I$ and $x'_J \subset x_J$). But this is not true because $c_\ell[\{x_i\} \cup x'_J]$ does not satisfy G-bijection.

B. A sufficient condition for decomposing into a tree of binary constraints

A partial presence of G-bijection allows decomposition. Here, we show that the decomposition is also allowed when G-bijection is totally present in a constraint c_ℓ and $| x_I |= 1$.

Theorem 3: Given a constraint c_ℓ of arbitrary arity, if there are two disjoint subsets of variables x_I and x_J with $S(c_\ell) = x_I \cup x_J$ and $| x_I |= 1$, $c_\ell[x_I \cup x_J]$ satisfies G-bijection, so c_ℓ can be decomposed into a tree of binary constraints connecting the variable in x_I to each variable in x_J without loss of equivalence.

Proof: In order to prove that equivalence is preserved after decomposition, it suffices to show that any assignment $\mathcal{A}$ satisfies $R(c_\ell)$ iff it does not violate any new binary constraint.

($\Longleftarrow$) Suppose, for a contradiction, that there is an assignment $\mathcal{A}$ which does not violate any new constraint (obtained after decomposing c_ℓ) but does not satisfy the original constraint c_ℓ. This case is only possible if there are at least

- two tuples t'_i, t''_i in $R(c_\ell)$ and
- a variable $x_j \in x_J$

s.t.:

- $t'_i[\{x_i\}] = \mathcal{A}[\{x_i\}]$ and $\forall x_k \in x_J \setminus \{x_j\}$, $t'_i[\{x_k\}] = \mathcal{A}[\{x_k\}]$ and $t'_i[\{x_j\}] \neq \mathcal{A}[\{x_j\}]$
- $t''_i[\{x_i\}] = \mathcal{A}[\{x_i\}]$ and $t''_i[\{x_j\}] = \mathcal{A}[\{x_j\}]$ and $\exists x_l \in x_J \setminus \{x_j\} \mid t''_i[\{x_l\}] \neq \mathcal{A}[\{x_l\}]$

Thus, after decomposing c_ℓ, $\mathcal{A}$ will not violate any new constraint whereas initially it does not satisfy the original constraint c_ℓ.

This case cannot exist because by definition of bijection, each value in $D(x_i)$ must be compatible with only one v_J in $R(c_\ell)[x_J]$. This provides the contradiction we were looking for.

($\Longrightarrow$) It is clear that if an assignment $\mathcal{A}$ is consistent for $R(c_\ell)$ then so is for the new binary constraints obtained after decomposition. In fact, each value in v_J may be compatible with more than one value v_i in $D(x_i)$, but v_i is compatible with exactly one value of each variable in x_J. Thus the set of all compatible values with a value v_i in $D(x_i)$ constitute a unique v_J in $R(c_\ell)[x_J]$. □

If we consider $x_I = \{x_i\}$ and $x_J = \{x_j, x_m, x_k\}$, the constraint of Example 1 satisfies the condition of the previous theorem and can also be decomposed into three binary constraints $c_{i,j}$, $c_{i,m}$ and $c_{i,k}$.

C. A sufficient condition for decomposing into a chain of binary constraints

Partial or total presence of (G-)bijection is a sufficient condition to a given constraint to be decomposable. Here we will add a weaker condition to ensure the decomposition of non-binary constraint into a chain of binary ones. Firstly we start by the following definition:

Definition 5 (BCC): Given a non-binary constraint c_ℓ, a binary constraint chain of c_ℓ, denoted BCC(c_ℓ), is a set of binary constraints $c_{i,i+1}$ with $1 \leq i < a_\ell$ and $x_i, x_{i+1} \in S(c_\ell)$.

It is clear that binary constraint chain of a given constraint c_ℓ is not one-off. For each variable ordering, we obtain a new BCC. For Example 2, $\{c_{1,2}, c_{2,3}, c_{3,4}, c_{4,5}\}$ is a possible BCC(c_ℓ).

Proposition 1: Given a constraint c_ℓ of arbitrary arity, if for all three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$, $c_\ell[x_I \cup x_J]$ satisfies G-bijection, so c_ℓ can be decomposed into BCC(c_ℓ) without loss of equivalence.

Proof: Let c_ℓ be a constraint of arbitrary arity s.t. for all three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$, $c_\ell[x_I \cup x_J]$ satisfies G-bijection. We will prove by contradiction that replacing c_ℓ by one of its BCC preserves equivalence. For this, we suppose that there is an assignment $\mathcal{A}$ which does not violate any binary constraint in BCC(c_ℓ) but does not satisfy the original constraint c_ℓ. We can express it as follows:

- $\forall i$ with $1 \le i < a_\ell$, $\mathcal{A}[\{x_i, x_{i+1}\}] \in R(c_{i,i+1})$ but
- $\mathcal{A} \notin R(c_\ell)$

So, there is at least a value $v_i \in \mathcal{A}$ which is not compatible with $v_j \in \mathcal{A}$ (we indicate that j is equal to either $i-1$ or $i+1$). We necessarily have two tuples $t_\ell, t'_\ell \in R(c_\ell)$ s.t. neither t_ℓ or t'_ℓ is equal to $\mathcal{A}$, i.e.:

- $\forall x_m \in S(c_\ell)\backslash\{x_j\}$ $t_\ell[\{x_m\}] = \mathcal{A}[\{x_m\}]$ and $t_\ell[\{x_j\}] \neq \mathcal{A}[\{x_j\}]$, with $t_\ell[\{x_j\}] = v'_j$
- $t'_\ell[\{x_i\}] = \mathcal{A}[\{x_i\}]$ and $\exists x_j, x_k \in S(c_\ell) \setminus \{x_i\} \mid t'_\ell[\{x_j\}] = \mathcal{A}[\{x_j\}]$ and $t'_\ell[\{x_k\}] \neq \mathcal{A}[\{x_k\}]$

This implies that (v_i, v_j) and $(v_i, v'_j) \in R(c_\ell)$. In this case, if we denote $x_I = x_i$, $x_J = x_j$ and $x_K = S(c_\ell) \setminus \{x_i, x_j\}$, c_ℓ does not satisfy G-bijection whatever x_I, x_J and x_k which contradicts our assumption. □

Example 3: Consider the ternary constraint c_ℓ which satisfies G-bijection whatever x_I and x_J.

$R(c_\ell)$		
x_i	x_j	x_k
a	b'	c'
a''	b	c''
a'	b''	c

$R(c_{i,j})$	$R(c_{j,k})$
$x_i\ x_j$	$x_j\ x_k$
$a\ b'$	$b'\ c'$
$a''\ b$	$b\ c''$
$a'\ b''$	$b''\ c$

D. Link with general multivalued dependency (GMvD)

Let's establish the link with general multivalued dependency[1] [12].

Proposition 2: Considering a constraint c_ℓ of arbitrary arity and three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$, if $c_\ell[x_I \cup x_J]$ satisfies G-bijection then c_ℓ satisfies GMvD.

Proof: Suppose for a given constraint c_ℓ and three disjoint subsets x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$ and $c_\ell[x_I \cup x_J]$ satisfies G-bijection but c_ℓ does not satisfy GMvD. There are $v_I \in R(c_\ell)$, $v_J, v'_j \in R(c_\ell)$ and $v_K, v'_K \in R(c_\ell)$ s.t.

- $(v_I, v_J, v_K) \in R(c_\ell)$
- $(v_I, v'_J, v'_K) \in R(c_\ell)$
- $(v_I, v_J, v'_K) \notin R(c_\ell)$
- $(v_I, v'_J, v_K) \notin R(c_\ell)$

We supposed that $c_\ell[x_I \cup x_J]$ satisfies G-bijection, so there is a unique value $v_J \in R(c_\ell)[x_J]$ such as $(v_I, v_J) \in R(c_\ell)[x_I \cup x_J]$ which is not actually the case. □

The converse is false by means of example 4.

Example 4: Consider the ternary constraint c_ℓ which satisfies MvD ($x_i \twoheadrightarrow x_j, x_k$ and $x_k \twoheadrightarrow x_i, x_j$) but is not a bijection.

$R(c_\ell)$		
x_i	x_j	x_k
a	b	c
a	b'	c'
a'	b	c'
a	b	c'
a	b'	c

IV. Decomposition based on absence of Dual broken triangle

We now turn to the second tractable class namely DBTP [5] which represents an extension of BTP to CSPs of arbitrary arity. We will begin with the following notation:

Notation 4: Given a constraint c_ℓ of arbitrary arity, we denote by $Inst(c_\ell) = (X', D', C')$ a binary sub-CSP with:

- $X' = \{x_i \in S(c_\ell)\}$
- $D' = \{D(x_i) \mid x_i \in X'\}$
- $C' = \{c_{i,j} \mid x_i, x_j \in S(c_\ell)$ and $R(c_{i,j}) = R(c_\ell)[\{x_i, x_j\}]\}$

We also denote by $Inst(x_I, x_J, x_K) = (X'', D'', C'')$ a sub-CSP with:

- $X'' = \{x_i \in x_I \cup x_J \cup x_K\}$
- $D'' = \{D(x_i) \mid x_i \in X''\}$
- $C'' = \{c_I, c_J$ and $c_K \mid \forall S = I, J$ or K, $S(c_S) = x_S$ and $R(c_S) = R(c_\ell)[x_S]\}$

In practice, if we have to check whether a tuple $t \in R(c_{i,j})$, we do not have to translate c_ℓ into $Inst(c_\ell)$, but we can directly use $c_\ell[\{x_i, x_j\}]$.

Theorem 4: Considering a constraint c_ℓ of arbitrary arity, if $Inst(c_\ell)$ does not contain any DBT, then c_ℓ can be replaced with binary constraints in $Inst(c_\ell)$ without loss of equivalence.

Proof: Let c_ℓ be a constraint of arbitrary arity s.t. $Inst(c_\ell)$ does not contain any DBT. We will prove by contradiction that replacing c_ℓ by $Inst(c_\ell)$ preserves equivalence. Obviously, we suppose that there is an assignment $\mathcal{A} = (..., v_i, ..., v_j, ..., v_k, ...)$ which does not violate any binary constraint in $Inst(c_\ell)$ but does not satisfy the original constraint c_ℓ. We can express it as follows:

- $\forall x_i, x_j \in S(c_\ell)$, $\mathcal{A}[\{x_i, x_j\}] \in R(c_{i,j})$ but
- $\mathcal{A} \notin R(c_\ell)$

So, there is at least a value $v_i \in \mathcal{A}$ which is not compatible with $v_j \in \mathcal{A}$. We necessarily have two tuples $t_\ell, t'_\ell \in R(c_\ell)$ s.t. neither t_ℓ or t'_ℓ is equal to $\mathcal{A}$, i.e.:

- $\forall x_m \in S(c_\ell)\backslash\{x_j\}$ $t_\ell[\{x_m\}] = \mathcal{A}[\{x_m\}]$ and $t_\ell[\{x_j\}] \neq \mathcal{A}[\{x_j\}]$, with $t_\ell[\{x_j\}] = v'_j$
- $t'_\ell[\{x_i\}] = \mathcal{A}[\{x_i\}]$ and $\exists x_j, x_k \in S(c_\ell) \setminus \{x_i\} \mid t'_\ell[\{x_j\}] = \mathcal{A}[\{x_j\}]$ and $t'_\ell[\{x_k\}] \neq \mathcal{A}[\{x_k\}]$ with $t_\ell[\{x_k\}] = v'_k$

In this way, the four couples (v_i, v'_j), (v_i, v'_k), (v'_j, v_k) and (v_j, v'_k) form a DBT on $c_{j,k}$. But we supposed that there is no DBT in $Inst(c_\ell)$. □

Going back to Example 4, $Inst(c_\ell)$ contains many DBT (e.g. $((a, b'), (a, c), (b, c), (a', b))$) so c_ℓ is not decomposable.

[1]Given a constraint c_ℓ of arbitrary arity, we say that c_ℓ satisfies general multivalued dependency (GMvD) if there are three disjoint subsets x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$ s.t. $\forall v_I \in R(c_\ell)[x_I], \forall v_J, v'_J \in R(c_\ell)[x_J]$ and $\forall v_K, v'_K \in R(c_\ell)[x_K]$ if

- $(v_I, v_J, v_K) \in R(c_\ell)$
- $(v_I, v'_J, v'_K) \in R(c_\ell)$

Then,

- $(v_I, v'_J, v_K) \in R(c_\ell)$ and
- $(v_I, v_J, v'_K) \in R(c_\ell)$

In this case, we denote $x_I \twoheadrightarrow x_J, x_K$.

The previous theorem can also be applied to decompose a non-binary constraint into three constraints of arity less than that of original one.

Corollary 1: Considering a constraint c_ℓ of arbitrary arity, if there are three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$ and $Inst(x_I, x_J, x_K)$ does not contain any DBT, then c_ℓ can be decomposed into three constraints $c_\ell[x_I \cup x_J]$, $c_\ell[x_I \cup x_K]$ and $c_\ell[x_J \cup x_K]$ without loss of equivalence.

A. Link with general interdependency (GID)

We now establish the link with a previous works, namely general interdependency[2] [12].

Proposition 3: Considering a constraint c_ℓ of arbitrary arity and three disjoint subsets of variables x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$, if $Inst(x_I, x_J, x_K)$ does not contain any DBT then c_ℓ satisfies GID.

Proof: Suppose for a given constraint c_ℓ and three disjoint subsets x_I, x_J and x_K with $S(C_\ell) = x_I \cup x_J \cup x_K$ and $Inst(x_I, x_J, x_K)$ does not contain any DBT but c_ℓ does not satisfy GID. There are $v_I, v'_I \in R(c_\ell)$, $v_J, v'_J \in R(c_\ell)$ and $v_K, v'_K \in R(c_\ell)$ s.t.

- $(v_I, v_J, v_K) \in R(c_\ell)$
- $(v_I, v'_J, v'_K) \in R(c_\ell)$
- $(v'_I, v_J, v'_K) \in R(c_\ell)$

We supposed that $Inst(x_I, x_J, x_K)$ does not contain any DBT, so c_ℓ is decomposable into three constraints $c_\ell[x_I \cup x_J]$, $c_\ell[x_I \cup x_K]$ and $c_\ell[x_J \cup x_K]$ without loss of equivalence. In this case, we have many DBT like $((v_J, v_K), (v_I, v_J), (v_I, v'_K), (v'_J, v'_K))$, which contradicts our assumption. □

The natural question arises: is the converse assertion true as well? By means of example 5, we can answer no because the contraint c_ℓ satisfies GID but $Inst(c_\ell)$ contains a DBT $((b, c'), (a, b), (a, c), (b', c))$ on $c_{j,k}$.

Example 5: Consider the ternary constraint c_ℓ which satisfies GID but $Inst(c_\ell)$ contains a DBT.

$R(c_\ell)$		
x_i	x_j	x_k
a	b	c
a'	b	c'
a''	b'	c

V. Conclusion and perspectives

It is a classical fact that instances of CSP can be converted in polytime to equivalent binary ones. There are several such reductions: dual encoding, hidden variable transformation and double encoding. A second way is to convert each non-binary constraint into a clique of binary ones, but unfortunately it does not preserve equivalence.

In this paper, we proposed to replace non-binary constraints by conjunctions of binary projections of the original constraint. We provided some conditions based on properties of tractable classes which ensure that translation of non-binary constraint into binary ones preserves equivalence. The first proof of decomposability relies on the presence of bijection while the second is based on the absence of DBT.

There are many directions to continue this work. First, it would be interesting to validate this study through an experimental trials. The second, it would be desirable to extend our theoretical results to other tractable properties. In other words, we have to check if other tractable properties could be used to check the decomposability of constraints or more generally to check its applicability in other subdomain in CP.

VI. Acknowledgment

I would like to thank Manel Slokom from LARODEC - Tunis/Tunisia for its help in writing this paper.

References

[1] Eugene C. Freuder. A Sufficient Condition for Backtrack-Free Search. *JACM*, 29 (1):24–32, 1982.

[2] Philippe David. When functional and bijective constraints make a CSP polynomial. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pages 224–231, 1993.

[3] Yuanlin Zhang, Roland H. C. Yap, and Joxan Jaffar. Functional elimination and 0/1/all constraints. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, July 18-22, 1999*, pages 175–180, 1999.

[4] Martin C. Cooper, Peter Jeavons, and Andras Salamon. Generalizing constraint satisfaction on trees: hybrid tractability and variable elimination. *Artificial Intelligence*, 174:570–584, 2010.

[5] Achref El Mouelhi, Philippe Jégou, and Cyril Terrioux. A Hybrid Tractable Class for Non-Binary CSPs. *Constraints*, 20(4):383–413, 2015.

[6] Martin C. Cooper, Philippe Jégou, and Cyril Terrioux. A microstructure-based family of tractable classes for csps. In *Principles and Practice of Constraint Programming - 21st International Conference, CP, 2015, Proceedings*, pages 74–88, 2015.

[7] Wady Naanaa. Extending the broken triangle property tractable class of binary csps. In *Proceedings of the 9th Hellenic Conference on Artificial Intelligence, SETN, 2016*, pages 3:1–3:6, 2016.

[8] Martin C. Cooper, Achref El Mouelhi, and Cyril Terrioux. Extending broken triangles and enhanced value-merging. In *Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Proceedings*, pages 173–188, 2016.

[9] Achref El Mouelhi, Philippe Jégou, and Cyril Terrioux. Hidden Tractable Classes: From Theory to Practice. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI, 2014*, pages 437–445, 2014.

[10] Martin C. Cooper, Aymeric Duchein, Achref El Mouelhi, Guillaume Escamocher, Cyril Terrioux, and Bruno Zanuttini. Broken triangles: From value merging to a tractable class of general-arity constraint satisfaction problems. *Artificial Intelligence*, 234:196 – 218, 2016.

[11] Achref El Mouelhi, Philippe Jégou, and Cyril Terrioux. Microstructures for csps with constraints of arbitrary arity. In *Proceedings of the Tenth Symposium on Abstraction, Reformulation, and Approximation, SARA 2013, 11-12 July 2013*, 2013.

[12] Achref El Mouelhi. On the decomposition of non-binary constraint into equivalent binary constraints. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, November 9-11, 2015*, pages 9–16, 2015.

[2] Given a constraint c_ℓ of arbitrary arity, we say that c_ℓ satisfies general interdependency (GID) if there are three disjoint subsets x_I, x_J and x_K with $S(c_\ell) = x_I \cup x_J \cup x_K$ and if $\forall v_I \in R(c_\ell)[x_I], \forall v_J, v'_J \in R(c_\ell)[x_J]$ and $\forall v_K, v'_K \in R(c_\ell)[x_K]$ s.t.

- $(v_I, v_J, v_K) \in R(c_\ell)$
- $(v_I, v'_J, v'_K) \in R(c_\ell)$

then, there is no $v'_I \in R(c_\ell)[x_I]$ s.t.

- $(v'_I, v'_J, v_K) \in R(c_\ell)$ or
- $(v'_I, v_J, v'_K) \in R(c_\ell)$