A BTP-Based Family of Variable Elimination Rules for Binary CSPs

Achref Mouelhi

A BTP-Based Family of Variable Elimination Rules for Binary CSPs

Achref El Mouelhi
Aix Marseille Univ, Université de Toulon
CNRS, ENSAM, LSIS
Marseille, France
achref.elmouelhi@lsis.org

Abstract

The study of broken-triangles is becoming increasingly ambitious, by both solving constraint satisfaction problems (CSPs) in polynomial time and reducing search space size through value merging or variable elimination. Considerable progress has been made in extending this important concept, such as dual broken-triangle and weakly broken-triangle, in order to maximize the number of captured tractable CSP instances and/or the number of merged values. Specifically, m-wBTP allows to merge more values than BTP. k-BTP, WBTP and m-BTP permit to capture more tractable instances than BTP.

Here, we introduce a new weaker form of BTP, which will be called m-fBTP for flexible broken-triangle property. m-fBTP allows on the one hand to eliminate more variables than BTP while preserving satisfiability and on the other to define new bigger tractable class for which arc consistency is a decision procedure. Likewise, m-fBTP permits to merge more values than BTP but less than m-wBTP.

1 Introduction

A wide range of real-life problems issue from Artificial Intelligence (AI) and Operational Research (OR) can be expressed as *Constraint Satisfaction Problems* (CSPs (Montanari 1974)). A constraint network consists of a set of *variables* X, each one has a finite set of *values* called *domain* D, and a finite set of *constraints* C. Each constraint is defined over a set of variables and represents a set of valid (or invalid) assignment of values to variables involved by the constraint. A *solution* to a CSP is an assignment of values to each variable which satisfies all the constraints. Checking whether a given CSP instance has a solution is known to be NP-complete. In general, the main techniques to achieve this task are based on backtracking algorithms, whose worst-case time complexity is $O(ed^n)$ where e, n and d are the number of constraints, the number of variables and the maximum domain size, respectively. In order to reduce this exponential time complexity, many different approaches have been proposed. The first one, called filtering by consistency, consists in removing inconsistent values (Mackworth 1977). This approach leaves the set of solutions unchanged. The second relies on merging values, satisfying some conditions, without affecting the existence of a solution (Freuder 1991; Likitvivatanavong and Yap 2013; Cooper et al. 2014; 2016). The last eliminates variables (Cohen et al. 2013; 2015) or constraints (Dechter and Dechter 1987) while preserving the satisfiability of the instance.

In a somewhat orthogonal direction, much research has been devoted to identifying *tractable classes*. In the litterature, several tractable classes have been defined but the *broken-triangle property* (BTP (Cooper, Jeavons, and Salamon 2008; 2010)) still remains at the heart of this research area. This property has some interesting characteristics from a solving viewpoint as well as reduction operations viewpoint. Indeed, BTP is not only defined for solving CSP in polytime, but also for reducing the size of CSP instances while preserving satisfiability. Specifically, the absence of broken-triangles has led, under some conditions, to variable elimination or domain reduction by merging domain values while preserving satisfiability. More recently, it has been proved that the presence of certain broken-triangles does not necessarily preclude defining tractable classes (Naanaa 2013; 2016; El Mouelhi, Jégou, and Terrioux 2014; 2015; Jégou and Terrioux 2015; Cooper, Jégou, and Terrioux 2015; Cooper, El Mouelhi, and Terrioux 2016) and/or merging values (Cooper, El Mouelhi, and Terrioux 2016). For example, k-BTP authorizes some broken-triangles and defines larger tractable classes than BTP but does not permit value merging. Likewise, m-wBTP does not forbid all broken-triangles and defines a maximal value-merging condition. Unfortunately, none of them allow variable elimination (see (Cooper, El Mouelhi, and Terrioux 2016)) although the initial definition of BTP permits it (Cohen et al. 2015). Moreover, k-BTP seems to be unusable beyond $k = 3$ and m-wBTP appears to be inexploitable when $m > 2$.

The main contribution of this work is providing a new weaker-form of BTP, called m-fBTP, which allows variable elimination and defines a new tractable class for which arc consistency is a decision procedure. The results proven in this paper also provide theoretical insight into the relationship between m-fBTP and some others previous extension of BTP such as k-BTP, m-wBTP, WBTP (Naanaa 2016).

So, our paper will be organised as follows: Section 2 recalls some definitions and notations. In section 3, we introduce the flexible broken-triangle property. Next, we show that m-fBTP is a maximal variable elimination condition. Section 4 defines a family of tractable classes based on m-fBTP. In section 5, we compare m-fBTP to some known tractable classes like k-BTP and WBTP. Finally we give a discussion and perspective for future work.

2 Formal background

Constraint satisfaction problems constitute an important tool for modeling and solving many different practical problems in Artificial Intelligence and Operations Research. In this paper, we consider only binary CSP instances, defined formally as follows:

Definition 1 (CSP). *An input binary CSP instance is a couple $I = (X, C)$, with*

- $X = \{x_1, ..., x_n\}$*: a set of n **variables**. each variable x_i has a **domain** $D(x_i)$ containing at most d values.*
- C*: a set of e binary **constraints**. Each constraint c_{ij} is a couple $(Scp(c_{ij}), Rel(c_{ij}))$ where:*
  - $Scp(c_{ij}) = \{x_i, x_j\} \subseteq X$*, is the **scope** of c_{ij},*
  - $Rel(c_{ij}) \subseteq D(x_i) \times D(x_j)$*, is the compatibility **relation**.*

If the constraint C_{ij} is not defined in C, then we consider C_{ij} to be a universal constraint (i.e. such that $Rel(C_{ij}) = D(x_i) \times D(x_j)$).

The output is called *solution*, i.e. an *assignment* of values to each variable in X which satisfies all constraints in C. A *partial solution* is an assignment $\mathcal{A} = (v_{\ell_1}, \ldots, v_{\ell_m}) \in D(x_{\ell_1}) \times \ldots \times D(x_{\ell_m})$ which satisfies all constraints C_{ij} such that $\{x_i, x_j\} \subseteq \{x_{\ell_1}, \ldots, x_{\ell_m}\}$.

Given a binary instance I, deciding whether I has a solution is well known to be NP-complete even for binary CSPs. Neverthless, there are some cases for which solving can be realized in polynomial time. In this case we speak about *tractable* classes. For example, BTP (for *broken-triangle Property* (Cooper, Jeavons, and Salamon 2010)) represents an important tractable class from a solving viewpoint as well as reduction operations viewpoint. The broken-triangle Property requires the *absence* of broken-triangles with respect to a given variable ordering. Formally, BTP is defined as follows:

Definition 2 (Broken-Triangle Property (Cooper, Jeavons, and Salamon 2010)). *Given a binary CSP instance I with a variable order $<$. A pair of values $v'_k, v''_k \in D(x_k)$ satisfies **BTP** if, for each pair of variables (x_i, x_j) such that $x_i < x_j < x_k$, $\forall v_i \in D(x_i)$, $\forall v_j \in D(x_j)$, if $(v_i, v_j) \in Rel(C_{ij})$, $(v_i, v'_k) \in Rel(C_{ik})$ and $(v_j, v''_k) \in Rel(C_{jk})$, then either $(v_i, v''_k) \in Rel(C_{ik})$ or $(v_j, v'_k) \in Rel(C_{jk})$.*
If $(v_i, v''_k) \notin Rel(C_{ik})$ and $(v_j, v'_k) \notin Rel(C_{jk})$, we say that (v'_k, v_i, v_j, v''_k) constitute a broken-triangle *on the values v'_k and v''_k. A variable x_k satisfies BTP if each pair of values in $D(x_k)$ satisfies BTP. An instance satisfies BTP if all its variables satisfy BTP.*

Graphically, BTP can be represented in the microstructure[1] (Jégou 1993) of I as shown in Figure 1. In the following, the absence of edge and the dotted edge will be used to represent forbidden tuples (tuples which violate the constraint).

[1]Given a binary CSP instance $I = (X, C)$, the *micro-structure* of I is the undirected graph $\mu(I) = (V, E)$ with:

- $V = \{(x_i, v_i) : x_i \in X, v_i \in D(x_i)\}$,
- $E = \{ \{(x_i, v_i), (x_j, v_j)\} \mid i \neq j, C_{ij} \notin C$ or $C_{ij} \in C, (v_i, v_j) \in Rel(C_{ij})\}$

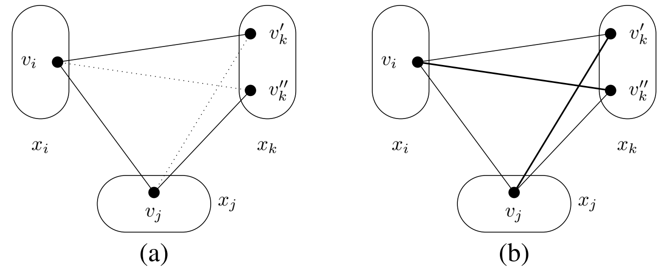


Figure 1: The assignments (v'_k, v_i, v_j, v''_k) form a broken-triangle in (a) but do not in (b) thanks to edges between (v_i, v''_k) and (v_j, v'_k).

Considering the variable ordering $x_i < x_j < x_k$, the CSP instance of Figure 1(a) is not BTP because of the incompatibility of (v_j, v'_k) and (v_i, v''_k). Thus, (v_i, v_j, v'_k, v''_k) constitute a broken-triangle. In Figure 1(b), $(v_i, v''_k) \in Rel(C_{ik})$ and $(v_j, v'_k) \in Rel(C_{jk})$, then the BTP is satisfied.

We now define value merging and variable elimination operations.

Definition 3. *(Cooper et al. 2016) **Merging** the values $v'_k, v''_k \in D(x_k)$ in a binary CSP instance I consists of replacing $v'_k, v''_k \in D(x_k)$ by a new value v_k which is compatible with all values which are compatible with either v'_k or v''_k. A **value-merging condition** is a polytime-computable property such that when it holds on a pair of values $v'_k, v''_k \in D(x_k)$, the instance obtained after merging the values v'_k and v''_k is satisfiable iff I was satisfiable.*

Definition 4. *(Cohen et al. 2015) **Eliminating** a variable x_k in a binary instance $I = (X, C)$ consists in replacing X by $X \setminus \{x_k\}$ and C by $C \setminus \{C_{ik} \in C \mid i \neq k\}$. A **variable-elimination condition** is a polytime-computable property such that when it holds on a variable x_k, the instance obtained after eliminating x_k is satisfiable iff I was satisfiable.*

In (Cohen et al. 2015), it has been shown that if there is no broken-triangles on each pair of values of a given variable x_k in an arc-consistent binary CSP instance I, then x_k can be eliminated from I without changing the satisfiability. Next, (Cooper et al. 2016) proved that even when this rule cannot be applied because of the presence of some broken-triangles, it is possible that there is a pair of values (v'_k, v''_k) in $D(x_k)$ which satisfies BTP. In this case, these two values are mergeable. For example, in Figure 1(b), the values v'_k and v''_k are mergeable. More recently, (Cooper, El Mouelhi, and Terrioux 2016) showed that even when some broken-triangles are present on a pair of values (v'_k, v''_k) which satisfies m-wBTP, merging v'_k and v''_k does not affect the satisfiability. Formally, m-wBTP is defined as follows:

Definition 5. *A pair of values $v'_k, v''_k \in D(x_k)$ satisfies m**-wBTP** where $m \leq n - 3$ if for each broken-triangle (v'_k, v_i, v_j, v''_k) with $v_i \in D(x_i)$ and $v_j \in D(x_j)$, there is a set of $r \leq m$ **support variables** $\{x_{\ell_1}, \ldots, x_{\ell_r}\} \subseteq X \setminus \{x_i, x_j, x_k\}$ such that for all $(v_{\ell_1}, \ldots, v_{\ell_r}) \in D(x_{\ell_1}) \times \ldots \times D(x_{\ell_r})$, if $(v_{\ell_1}, \ldots, v_{\ell_r}, v_i, v_j)$ is a partial solution, then there is $\alpha \in \{1, \ldots, r\}$ such that $(v_{\ell_\alpha}, v'_k), (v_{\ell_\alpha}, v''_k) \notin Rel(C_{\ell_\alpha k})$.*

Graphically, this definition can be represented through the micro-structure graph of Figure 2. The pair (v'_k, v''_k) satisfies 1-wBTP because the value v_ℓ in $D(x_\ell)$ is compatible with both v_i and v_j but is not with v'_k and v''_k. So we say that the assignments (v'_k, v_i, v_j, v''_k) forms a *weakly* broken-triangle which is supported by x_ℓ.

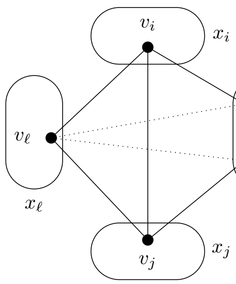


Figure 2: A weakly broken-triangle (v'_k, v_i, v_j, v''_k) since $(v'_k, v_\ell), (v''_k, v_\ell) \notin Rel(C_{k\ell})$.

Contrary to BTP, m-wBTP does not allow variable elimination (see Section 5 in (Cooper, El Mouelhi, and Terrioux 2016)). So, next section introduces the *flexible* broken-triangle concept which allows merging value and variable elimination while preserving satisfiability.

3 Flexible broken-triangles

A total absence of broken-triangles on a given variable in an arc-consistent CSP instance allows us to eliminate it without changing the satisfiability of the initial instance. In contrast, a total absence of weakly broken-triangles does not permit variable elimination. In practice, we have many examples of variables which can be eliminated despite the presence of certain broken-triangles while preserving satisfiability.

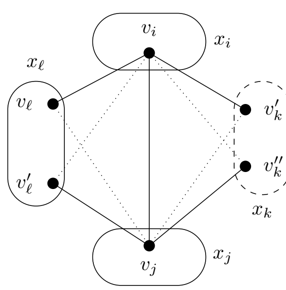


Figure 3: The dashed variable x_k can be eliminated despite the presence of a broken-triangle.

As shown in the inconsistent CSP instance of Figure 3, there is a broken-triangle on v'_k and v''_k, but after eliminating x_k this CSP instance still remains inconsistent. So, the presence of some broken-triangle on a given variable does not preclude variable elimination while preserving satisfiability. For this, we introduce the *flexible broken-triangles*.

Like m-wBTP, the m-fBTP is based on the concept of *support variables*. From a micro-structure viewpoint, these variables prevent the emergence of a new clique[2] (corresponds to a partial solution) which was not previously.

We begin by formally defining m-fBTP.

Definition 6. *A pair of values $v'_k, v''_k \in D(x_k)$ satisfies m-**fBTP** where $m \leq n-3$ if for each broken-triangle (v'_k, v_i, v_j, v''_k) with $v_i \in D(x_i)$ and $v_j \in D(x_j)$, there is a set of $r \leq m$ **support variables** $\{x_{\ell_1}, \ldots, x_{\ell_r}\} \subseteq X \setminus \{x_i, x_j, x_k\}$ such that for all partial solution $(v_{\ell_1}, \ldots, v_{\ell_r}) \in D(x_{\ell_1}) \times \ldots \times D(x_{\ell_r})$, there is $\alpha \in \{1, \ldots, r\}$ such that if $(v_{\ell_\alpha}, v_i) \in Rel(C_{\ell_\alpha i})$, then $(v_{\ell_\alpha}, v_j) \notin Rel(C_{\ell_\alpha j})$. In this case, we say that (v'_k, v_i, v_j, v''_k) is a flexible broken-triangle. A variable $x_k \in X$ satisfies m-fBTP if each pair of values $v'_k, v''_k \in D(x_k)$ satisfies m-fBTP.*

In other words, there is at least one value belonging to each parial solution which cannot be compatible with both v_i and v_j at the same time. If there is no variable which satisfies the previous conditions, then we will say that (v'_k, v_i, v_j, v''_k) is a *purely broken-triangle*. For example, the assignments (v'_k, v_i, v_j, v''_k) in Figure 2 form a purely broken-triangle because $(v_i, v_\ell) \in Rel(C_{i\ell})$ and $(v_j, v_\ell) \in Rel(C_{i\ell})$. Consequently, the pair (v'_k, v''_k) does not satisfy 1-fBTP.

Three different configurations of Definition 6 are given in Figure 4. In (a), there is no partial solution on the set of variables $\{x_{\ell_\beta}, x_{\ell_\gamma}\}$. Hence v'_k, v''_k in $D(x_k)$ clearly satisfies 2-fBTP. In (b), the pair of values v'_k, v''_k in $D(x_k)$ satisfies 2-fBTP because for the two partial solutions $(v'_{\ell_\beta}, v'_{\ell_\gamma})$ and $(v''_{\ell_\beta}, v''_{\ell_\gamma})$, we have $(v'_{\ell_\gamma}, v_j) \notin Rel(C_{\ell_\gamma j})$ and $(v''_{\ell_\beta}, v_i) \notin Rel(C_{\ell_\beta i})$. In (c), the pair of values v'_k, v''_k in $D(x_k)$ also satisfies 2-fBTP because for the two partial solutions $(v'_{\ell_\beta}, v'_{\ell_\gamma})$ and $(v''_{\ell_\beta}, v''_{\ell_\gamma})$, we have $(v'_{\ell_\gamma}, v_j) \notin Rel(C_{\ell_\gamma j})$ and $(v''_{\ell_\gamma}, v_i) \notin Rel(C_{\ell_\gamma i})$.

One can observe that in Figure 4 (c), the variable x_{ℓ_γ} alone supports the broken-triangle (v'_k, v_i, v_j, v''_k), so we can deduce that x_{ℓ_γ} and x_{ℓ_β} together support it. In Figure 4 (a) and (b), x_{ℓ_γ} and x_{ℓ_β} together support the broken-triangle (v'_k, v_i, v_j, v''_k) none of them alone support it.

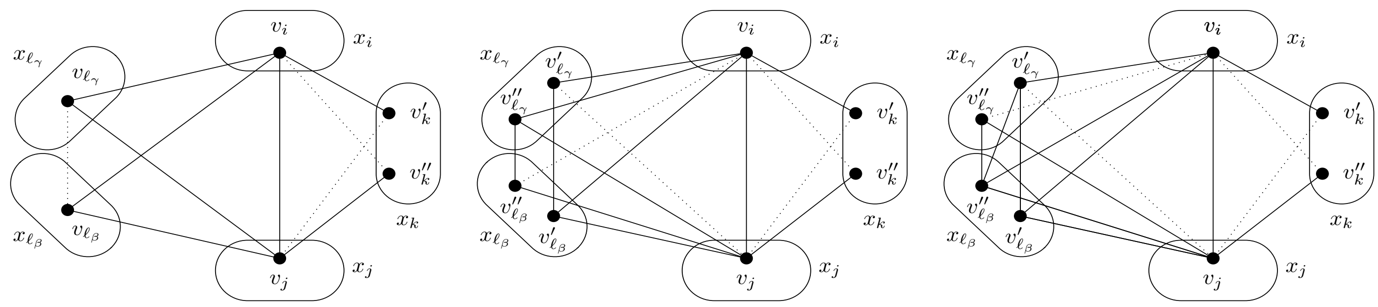
Proposition 1. *Given a binary CSP instance $I = (X, C)$, if a pair of values $v'_k, v''_k \in D(x_k)$ satisfies m-fBTP then it satisfies $(m+1)$-fBTP ($0 \leq m \leq n-4$).*

We now define the relationship between m-wBTP and m-fBTP.

Proposition 2. *In a binary CSP instance $I = (X, C)$, $\forall m,\ 0 \leq m \leq n-4$, if a pair $v'_k, v''_k \in D(x_k)$ satisfies m-fBTP, then it also satisfies m-wBTP.*

Proof. For each broken-triangle on v'_k, v''_k, there is a set of $r \leq m$ (with $0 \leq m \leq n-3$) support variables $\{x_{\ell_1}, \ldots, x_{\ell_r}\} \subseteq X \setminus \{x_i, x_j, x_k\}$ such that for all partial solution $(v_{\ell_1}, \ldots, v_{\ell_r}) \in D(x_{\ell_1}) \times \ldots \times D(x_{\ell_r})$, there is $\alpha \in \{1, \ldots, r\}$ such that if $(v_{\ell_\alpha}, v_i) \in Rel(C_{\ell_\alpha i})$, then $(v_{\ell_\alpha}, v_j) \notin Rel(C_{\ell_\alpha j})$. So each value v_{ℓ_α} in $D(x_{\ell_\alpha})$ cannot be compatible with v_i and v_j at the same time. Thus, there can be no partial solution $(v_{\ell_1}, \ldots, v_{\ell_r})$. As a result, the pair $v'_k, v''_k \in D(x_k)$ also satisfies m-wBTP. □

[2] A complete subgraph where each pair of vertices are connected.



(a) (b) (c)

Figure 4: Three different cases of two values v'_k and v''_k which satisfy 2-fBTP.

The converse is obiously false by means of Figure 2 where the pair (v'_k, v''_k) is 1-wBTP but is not 1-fBTP.

Corollary 1. *In a binary CSP instance $I = (X, C)$, merging a pair of values $v'_k, v''_k \in D(x_k)$ which satisfies 1-fBTP does not change the satisfiability of I.*

If we denote by m-fBTP-merging the merging operation based on m-fBTP, we can deduce that 0-wBTP-merging (Cooper, El Mouelhi, and Terrioux 2016) and 0-fBTP-merging correspond to BTP-merging defined in (Cooper et al. 2016) since they are based on zero support variables. Since BTP-merging generalises both neighbourhood substitution (Freuder 1991) and virtual interchangeability (Likitvivatanavong and Yap 2013) and m-fBTP-merging generalises BTP-merging for all $m \geq 0$, we immediately obtain the following result:

Corollary 2. *m-fBTP-merging generalises neighbourhood substitution and virtual interchangeability.*

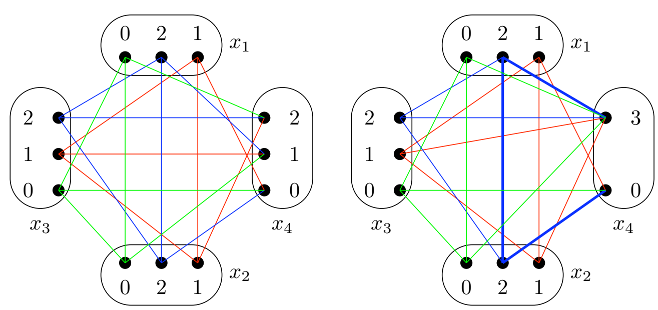
It is known that if a given variable x_k in an arc-consistent binary instance I satisfies BTP then x_k can be eliminated without modifying the satisfiability of I (Cohen et al. 2015). A similar result can also be shown for the variables satisfying m-fBTP. To do it, we should prove the following lemma:

Lemma 1. *Given a variable x_k which satisfies m-fBTP, after merging a pair of values $v''_k, v'''_k \in D(x_k)$ into a new value v'_k, no purely broken-triangle can appear on x_k.*

Proof. We assume, for a contradiction, that after merging a pair of values v''_k, v'''_k of a variable x_k which satisfies m-fBTP into a new value v'_k, we introduced a new purely broken-triangle (v_k, v_i, v_j, v'_k). So we have $(v_i, v_j) \in Rel(C_{ij})$ (1), $(v_i, v_k) \in Rel(C_{ik})$ (2), $(v_j, v'_k) \in Rel(C_{jk})$ (3), $(v_j, v_k) \notin Rel(C_{jk})$ (4) and $(v_i, v'_k) \notin Rel(C_{ik})$ (5). By definition 3, we obtain $(v_i, v''_k) \notin Rel(C_{ik})$ (a), $(v_i, v'''_k) \notin Rel(C_{ik})$ (b), and either $(v_j, v''_k) \in Rel(C_{jk})$ (c) or $(v_j, v'''_k) \in Rel(C_{jk})$ (d).

(2), (1), (c), (a), and (4) $\Rightarrow$ a broken-triangle (v_k, v_i, v_j, v''_k) and (2), (1), (d), (b) and (4) $\Rightarrow$ a broken-triangle (v_k, v_i, v_j, v'''_k). In both cases, we had at least one broken-triangle before merging v''_k and v'''_k. So, there is a set of $r \leq m$ support variables $\{x_{\ell_1}, \ldots, x_{\ell_r}\} \subseteq X \setminus \{x_i, x_j, x_k\}$ such that for all partial solution $(v_{\ell_1}, \ldots, v_{\ell_r}) \in D(x_{\ell_1}) \times \ldots \times D(x_{\ell_r})$, there is $\alpha \in \{1, \ldots, r\}$ such that if $(v_{\ell_\alpha}, v_i) \in Rel(C_{\ell_\alpha i})$, then $(v_{\ell_\alpha}, v_j) \notin Rel(C_{\ell_\alpha j})$. In this way, the set of r support variables $\{x_{\ell_1}, \ldots, x_{\ell_r}\}$ also support the broken-triangle (v_k, v_i, v_j, v'_k). Thus, (v_k, v_i, v_j, v'_k) is not a purely broken-triangle. But this contradicts our initial assumption. Finally, merging two values v''_k, v'''_k in the domain of a variable x_k which satisfies m-fBTP does not introduce a purely broken-triangle. □

Lemma 1 cannot be extended to all pair of values which satisfies m-wBTP (and does not satisfy m-fBTP (Cooper, El Mouelhi, and Terrioux 2016)). Indeed, Figure 5(a) illustrates the case of a variable x_4 which satisfies 1-wBTP since the variable x_3 supports all the broken-triangles on x_4. Figure 5(b) is obtained after merging the values 2 and 1 into a new value 3. Hence, the variable x_3 no longer support the broken-triangle $(3, 2, 2, 0)$ (in bold) because the value $2 \in D(x_3)$ is compatible at the same time with 2 ($\in D(x_1)$), 2 ($\in D(x_2)$) and 3 ($\in D(x_4)$).



(a) (b)

Figure 5: (a) A variable x_4 which satisfies 1-wBTP in an arc-consistent CSP instance. (b) The CSP instance obtained from I after merging the values 1 and 2 into a new value 3.

We now establish the link with variable elimination.

Theorem 1. *Given an arc-consistent CSP instance $I = (X, C)$, if a variable $x_k \in X$ satisfies m-fBTP, then it can be eliminated from I while preserving satisfiability.*

Proof. Given an arc-consistent CSP instance $I = (X, C)$ and a variable $x_k \in X$ which satisfies m-fBTP. Since value merging makes no empty domain, we will merge each pair of values in $D(x_k)$ until obtaining a unique value since merging a pair of values does not introduce a new purely broken-triangle on x_k (thanks to Lemma 1). As I is arc-consistent, so any consistent assignment $\mathcal{A}$ to $X \backslash \{x_k\}$ can be extended to x_k because $D(x_k)$ contains a unique value and each value $\mathcal{A}[x_i]$ has a support in $D(x_k)$. So, the unique value in $D(x_k)$ is compatible with each value in $\mathcal{A}$. Thus, x_k can be eliminated without changing the satisfiability of I. □

4 A maximal variable-elimination condition

It has been proved that the variable which satisfies BTP can be eliminated while preserving satisfiability (Cohen et al. 2015). In section 3, we have shown that even if a variable does not satisfy BTP it can be eliminated without changing the satisfiability of the instance while this variable satisfies m-fBTP. Thus, in an obvious sense, satisfying BTP is not a maximal variable-elimination condition.

Definition 7. *A variable-elimination condition is* maximal *if the elimination of any other variable not respecting the condition necessarily leads to a modification of the satisfiability of some instance.*

In this section, we show that m-fBTP is a maximal variable-elimination condition when $m = n - 3$.

Theorem 2. *In an unsatisfiable binary CSP instance $I = (X, C)$, there is no variable not satisfying m-fBTP for $m = n - 3$ and which can be eliminated while preserving satisfiability.*

Proof. Considering an unsatisfiable binary CSP instance $I = (X, C)$ and a variable x_k which does not satisfy m-fBTP for $m = n - 3$. By the definition of m-fBTP, there is a broken-triangle (v'_k, v_i, v_j, v''_k), with $v_i \in D(x_i)$, $v_j \in D(x_j)$ and $v'_k, v''_k \in D(x_k)$. And there is $(v_{\ell_1}, \dots, v_{\ell_m}) \in D(x_{\ell_1}) \times \dots \times D(x_{\ell_m})$, where $\{x_{\ell_1}, \dots, x_{\ell_m}\} = X \setminus \{x_i, x_j, x_k\}$, such that $(v_{\ell_1}, \dots, v_{\ell_m})$ is a partial solution and for all $\alpha \in \{1, \dots, m\}$ we have $(v_{\ell_\alpha}, v_i) \in Rel(C_{\ell_\alpha i})$ and $(v_{\ell_\alpha}, v_j) \in Rel(C_{\ell_\alpha j})$. In terms of micro-structure we have a $(n-1)$-clique (a subset of $n-1$ vertices that induces a complete subgraph) that we denote Cl.

We have a broken-triangle, and so: $(v_i, v''_k) \notin Rel(C_{ik})$, $(v_j, v'_k) \notin Rel(C_{jk})$, $(v_i, v'_k) \in Rel(C_{ik})$ and $(v_j, v''_k) \in Rel(C_{jk})$. After eliminating x_k, and by definition of elimination, the obtained instance I' has $(n-1)$ variables and its micro-structure contains the $(n-1)$-clique Cl. According to Property 2 in (Jégou 1993), Cl corresponds to a solution of I'. Thus, we introduced a solution which did not exist in the initial instance since $(v_i, v''_k) \notin Rel(C_{ik})$ and $(v_j, v'_k) \notin Rel(C_{jk})$. It follows that the elimination of variable which does not satisfy m-fBTP does not preserve satisfiability. □

We can now deduce the desired result.

Corollary 3. *$(n-3)$-fBTP is a maximal variable-elimination condition.*

5 Tractability of m-fBTP instances

Contrary to k-BTP and m-wBTP which sometimes need a high level of consistency, we show that arc consistency is a decision procedure for m-fBTP. We firstly start with extend m-fBTP definition to instances.

Definition 8. *A binary CSP instance I with a variable ordering $<$ satisfies m-fBTP relative to this order if for all variables x_k, each pair of values in $D(x_k)$ satisfies m-fBTP in the sub-instance of I on variables $x_i \le x_k$ ($m \le n-3$).*

We now prove that m-fBTP is conservative[3] (Cooper, Jeavons, and Salamon 2010), m-fBTP holds even after enforcing any filtering consistency which only removes values from domains.

Lemma 2. *m-fBTP with respect to any fixed variable ordering is conservative.*

Proof. It is clear that m-fBTP holds for a binary CSP instance thanks to the absence of some tuples. Obviously, removing values from the domain of any variable in a CSP instance cannot add new tuples. Thus, m-fBTP still holds. □

Theorem 3. *Arc consistency is a decision procedure for any binary CSP instance $I = (X, C)$ which satisfies m-fBTP ($1 \le m \le n-3$).*

Proof. Let $I = (X, C)$ be a binary CSP instance satisfying m-fBTP with respect to a variable ordering $<$. We begin by enforcing arc consistency. If this results to an empty domain, then obviously the obtained instance has no solution. Otherwise, thanks to Lemma 2, we know that the obtained instance will also satisfy m-fBTP. According to Theorem 1, we can proceed iteratively to eliminate the last variable with respect to $<$ until obtaining an instance with three variables x_1, x_2 and x_3. As I is becoming arc-consistent, so there is no empty domain. Hence, $D(x_1)$ (respectively $D(x_2)$) must contain at least a value v_1 (resp. v_2) such that $(v_1, v_2) \in Rel(C_{12})$ (1). We will suppose, for a contradiction, that the assignment $\mathcal{A} = (v_1, v_2)$ cannot be consistenly extended to x_3. For this, we assume that there is no $v_3 \in D(x_3)$ which is consistent with both v_1 and v_2. But, by arc consistency, we should have two values $v'_3, v''_3 \in D(x_3)$ such that $(v_1, v'_3) \in Rel(C_{13})$ (2) and $(v_2, v'_3) \in Rel(C_{23})$ (3). Note that v'_3 and v''_3 must be different and $(v_1, v''_3) \notin Rel(C_{13})$ (4) and $(v_2, v'_3) \notin Rel(C_{23})$ (5) (otherwise we contradicts our hypothesis).

In this way, (1), (2), (3), (4) and (5) form a purely broken-triangle on x_k which can be supported by no other variable. Indeed, by Definition 6, any variable x_ℓ must be different from $\{x_i, x_j, x_k\}$. Thus, this contradicts our assumption. Finally, $\mathcal{A}$ can be consistently extended to x_3. □

The following theorem is a logical consequence of Corollary 3 and Theorem 3.

Theorem 4. *The class of binary CSP instances which satisfy $(n-3)$-fBTP defines the biggest tractable class resolved by variable elimination.*

As with m-wBTP, checking whether it is possible to compute, in polytime, a variable ordering for which a binary CSP instance satisfies m-fBTP still remains an open question.

[3]A class Γ of CSP instances is said conservative with respect to a filtering consistency ϕ if it is closed under ϕ, that is, if the instance obtained after the application of ϕ still belongs to Γ.

6 m-fBTP vs some tractable classes based on BTP

We now compare m-fBTP to some tractable classes based on BTP. We start with k-BTP[4] (Cooper, Jégou, and Terrioux 2015).

Theorem 5. *m-fBTP and k-BTP are incomparable.*

Proof. Figure 2 shows a binary CSP instance which satisfies 3-BTP but does not satisfy 1-fBTP. Figure 6 illustrates the case of an instance which satisfies 1-fBTP but does not satisfy 3-BTP. Indeed, there are broken-triangles on the variable x_k for each pair of other variables, but in each case the fourth variable is a support variable. $\square$

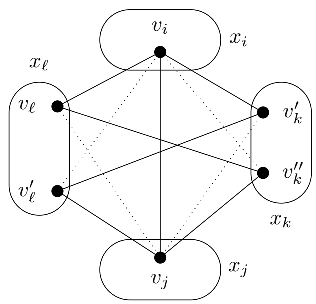


Figure 6: An instance which satisfies 1-fBTP but does not satisfy 3-BTP, for the variable ordering $x_\ell < x_i < x_j < x_k$.

We move to WBTP[5] (Naanaa 2016).

Theorem 6. *1-fBTP $\subsetneq$ WBTP.*

Proof. Obviously because both WBTP and 1-fBTP use a unique support variable and their condition depends only from v_i and v_j. $\square$

The converse is false by means of Figure 7(a). In fact, the binary CSP instance satisfies WBTP (both $x_{\ell_\beta}$and x_{ℓ_γ} support the broken-triangle (v'_k, v_i, v_j, v''_k)) but does not satisfy 1-fBTP.

Theorem 7. *2-fBTP and WBTP are incomparable.*

Proof. Figure 7(a) shows a binary CSP instance which satisfies WBTP but does not satisfy 2-fBTP. More precisely, $(v_{\ell_\beta}, v_{\ell_\gamma})$ is a partial solution but $(v_{\ell_\gamma}, v_j) \in Rel(C_{\ell_\gamma j})$, $(v_{\ell_\beta}, v_i) \in Rel(C_{\ell_\gamma i})$, $(v_{\ell_\gamma}, v_i) \in Rel(C_{\ell_\gamma i})$, $(v_{\ell_\beta}, v_j) \in Rel(C_{\ell_\gamma j})$. Figure 7(b) illustrates the case of an instance which satisfies 2-fBTP but does not satisfy WBTP because v''_{ℓ_γ} is compatible with v_i, v_j and v'_k, and v'_{ℓ_β} is compatible with v_i, v_j and v''_k. $\square$

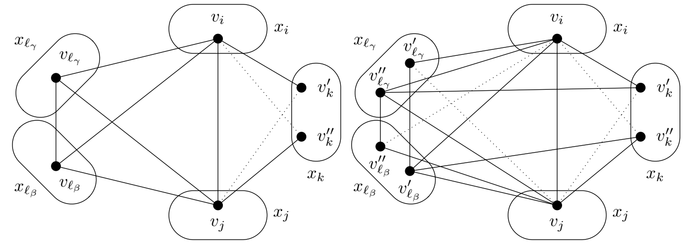


Figure 7: (a) A binary CSP instance which is WBTP but is not 2-fBTP. (b) A binary CSP instance which is 2-fBTP but is not WBTP.

Figure 8 summarizes some relation between tractable classes based on BTP. An arc from c_1 to c_2 (resp. a dashed line between c_1 and c_2) means that $c_1 \subsetneq c_2$ (resp. c_1 and c_2 are incomparable).

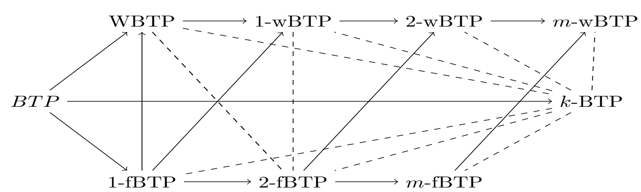


Figure 8: Relationship between tractable classes based on BTP.

7 Conclusion

BTP relies on absence of broken-triangle to define an important tractable class and to allow reducing search space size through value merging or variable elimination. Recently, many new weaker versions of BTP, which authorize the presence of some broken-triangle like k-BTP, WBTP and m-wBTP, have been studied but none of them define tractable class and permit variable elimination and value merging simultaneously. Moreover, much of these versions, except WBTP, require a high level of consistency. In this paper, we have proposed a new light version of BTP, called m-fBTP for flexible broken-triangle property. m-fBTP is based on support variable concept and permits to cover some imperfections of previous versions. More precisely, it allows value merging, represents a maximal variable-elimination condition and also defines a tractable class solved by arc consistency. m-fBTP is incomparable with the patterns described in (Cooper and Zivny 2016) and which characterise tractable classes for CSPs defined by partially-ordered forbidden patterns and solved by arc consistency. It would be interesting to generalise this family of definitions to non-binary CSPs.

Acknowledgments

I would like to thank Martin C. Cooper and Wady Naanaa for their useful comments during preparation of this work.

[4]A binary CSP instance I satisfies the k-BTP property for a given k $(2 \le k < n)$ relative to a variable order $<$ if, for all subsets of variables $x_{i_1}, x_{i_2}, \dots, x_{i_{k+1}}$ such that $x_{i_1} < x_{i_2} < \dots < x_{i_{k+1}}$, there is at least one pair of variables $(x_{i_j}, x_{i_{j'}})$ with $1 \le j < j' \le k$ such that there is no broken-triangle on x_{k+1} relative to x_{i_j} and $x_{i_{j'}}$.

[5]A binary CSP instance equipped with an order $<$ on its variables satisfies WBTP (Weak broken-triangle Property) if for each triple of variables $x_i < x_j < x_k$ and for all $v_i \in D(x_i)$, $v_j \in D(x_j)$ such that $(v_i, v_j) \in Rel(C_{ij})$, there is a variable $x_\ell < x_k$ such that when $v_\ell \in D(x_\ell)$ is compatible with v_i and v_j, then $\forall v_k \in D(x_k)$, if $(v_\ell, v_k) \in Rel(C_{\ell k})$ then $(v_i, v_k) \in Rel(C_{ik})$ and $(v_j, v_k) \in Rel(C_{jk})$

References

Cohen, D. A.; Cooper, M. C.; Escamocher, G.; and Zivny, S. 2013. Variable Elimination in Binary CSP via Forbidden Patterns. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, 2013*, 517–523.

Cohen, D. A.; Cooper, M. C.; Escamocher, G.; and Zivny, S. 2015. Variable and value elimination in binary constraint satisfaction via forbidden patterns. *J. Comput. Syst. Sci.* 81(7):1127–1143.

Cooper, M. C., and Zivny, S. 2016. The power of arc consistency for CSPs defined by partially-ordered forbidden patterns. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, 2016*, 652–661.

Cooper, M. C.; El Mouelhi, A.; Terrioux, C.; and Zanuttini, B. 2014. On Broken Triangles. In *Principles and Practice of Constraint Programming - 20th International Conference, CP 2014. Proceedings*, 9–24.

Cooper, M. C.; Duchein, A.; El Mouelhi, A.; Escamocher, G.; Terrioux, C.; and Zanuttini, B. 2016. Broken triangles: From value merging to a tractable class of general-arity constraint satisfaction problems. *Artificial Intelligence* 234:196 – 218.

Cooper, M. C.; El Mouelhi, A.; and Terrioux, C. 2016. Extending broken triangles and enhanced value-merging. In *Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Proceedings*, 173–188.

Cooper, M. C.; Jeavons, P.; and Salamon, A. 2008. Hybrid tractable CSPs which generalize tree structure. In *Proceedings of ECAI*, 530–534.

Cooper, M. C.; Jeavons, P.; and Salamon, A. 2010. Generalizing constraint satisfaction on trees: hybrid tractability and variable elimination. *Artificial Intelligence* 174:570–584.

Cooper, M. C.; Jégou, P.; and Terrioux, C. 2015. A microstructure-based family of tractable classes for CSPs. In *Principles and Practice of Constraint Programming - 21st International Conference, CP, 2015, Proceedings*, 74–88.

Dechter, A., and Dechter, R. 1987. Removing redundancies in constraint networks. In *Proceedings of the 6th National Conference on Artificial Intelligence.*, 105–109.

El Mouelhi, A.; Jégou, P.; and Terrioux, C. 2014. Hidden Tractable Classes: From Theory to Practice. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI, 2014*, 437–445.

El Mouelhi, A.; Jégou, P.; and Terrioux, C. 2015. A Hybrid Tractable Class for Non-Binary CSPs. *Constraints* 20(4):383–413.

Freuder, E. C. 1991. Eliminating interchangeable values in constraint satisfaction problems. In *Proceedings of the 9th National Conference on Artificial Intelligence, 1991, Volume 1.*, 227–233.

Jégou, P., and Terrioux, C. 2015. The extendable-triple property: A new CSP tractable class beyond BTP. In *Proceedings of AAAI*, 3746–3754.

Jégou, P. 1993. Decomposition of Domains Based on the Micro-Structure of Finite Constraint Satisfaction Problems. In *Proceedings of the 11th National Conference on Artificial Intelligence, 1993.*, 731–736.

Likitvivatanavong, C., and Yap, R. H. C. 2013. Many-to-many interchangeable sets of values in csps. In *Proceedings of SAC*, 86–91.

Mackworth, A. K. 1977. Consistency in Networks of Relations. *Artificial Intelligence* 8:99–118.

Montanari, U. 1974. Networks of Constraints: Fundamental Properties and Applications to Picture Processing. *Artificial Intelligence* 7:95–132.

Naanaa, W. 2013. Unifying and extending hybrid tractable classes of csps. *J. Exp. Theor. Artif. Intell.* 25(4):407–424.

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