Microstructures for CSPs with Constraints of Arbitrary Arity

Achref El Mouelhi and **Philippe Jégou** and **Cyril Terrioux**
LSIS - UMR CNRS 7296
Aix-Marseille Université
Avenue Escadrille Normandie-Niemen
13397 Marseille Cedex 20 (France)
{achref.elmouelhi, philippe.jegou, cyril.terrioux}@lsis.org

Abstract

Many works have studied the properties of CSPs which are based on the structures of constraint networks, or based on the features of compatibility relations. Studies on structures rely generally on properties of graphs for binary CSPs and on properties of hypergraphs for the general case, that is CSPs with constraints of arbitrary arity. In the second case, using the dual representation of hypergraphs, that is a reformulation of the instances, we can exploit notions and properties of graphs. For the studies of compatibility relations, the exploitation of properties of graphs is possible studying a graph called microstructure which allows to reformulate instances of binary CSP. Unfortunately, this approach is limited to CSPs with binary constraints.

In this paper, we propose theoretical tools based on graphs to represent microstructures for the general case. This approach avoids to exploit directly hypergraphs, even if the microstructure based on hypergraphs has already been mentioned in (Cohen 2003). The advantage of such an approach is that the literature of Graph Theory is really more extended than one of Hypergraph Theory. Thus the theoretical results and efficient algorithms are more numerous, offering a larger number of existing tools which can be operated. We introduce here three possible definitions of microstructures based on graphs. We show how these representations can form new theoretical tools to generalize a number of results already obtained on binary CSPs. We think that these representations should be of interest for the community, firstly for the generalization of existing results, but also to obtain original results for CSPs with constraints of arbitrary arity.

Preliminaries

Constraint Satisfaction Problems (CSPs, see (Rossi, van Beek, and Walsh 2006) for a state of the art) provide an efficient way of formulating problems in computer science, especially in Artificial Intelligence.

Formally, a *constraint satisfaction problem* is a triple (X, D, C), where $X = \{x_1, \dots, x_n\}$ is a set of n variables, $D = (D_{x_1}, \dots, D_{x_n})$ is a list of finite domains of values, one per variable, and $C = \{C_1, \dots, C_e\}$ is a finite set of e constraints. Each constraint C_i is a pair $(S(C_i), R(C_i))$, where $S(C_i) = \{x_{i_1}, \dots, x_{i_k}\} \subseteq X$ is the *scope* of C_i, and $R(C_i) \subseteq D_{x_{i_1}} \times \dots \times D_{x_{i_k}}$ is its *compatibility relation*. The *arity* of C_i is $|S(C_i)|$. We assume that each variable appears at least in the scope of one constraint and that the relations are represented in extension (e.g. by providing the list of allowed tuples). A CSP is called *binary* if all constraints are of arity 2 (we denote C_{ij} the binary constraint whose scope is $S(C_{ij}) = \{x_i, x_j\}$). Otherwise, if the constraints are of arbitrary arity, a CSP is said to be *n-ary*. The structure of the constraint network is represented by the hypergraph (X, C) (which is a graph in the binary case) whose vertices correspond to variables and edges to the constraint scopes. An assignment on a subset of X is said to be *consistent* if it does not violate any constraint. Testing whether a CSP has a *solution* (i.e. a consistent assignment on all the variables) is known to be NP-complete.

So, many works have been realized to make the solving of instances more efficient by using optimized backtracking algorithms, filtering techniques based on constraint propagation, heuristics ... Another way is related to the study of tractable classes defined by properties of constraint networks. E.g., it has been shown that if the structure of this network, that is a graph for binary CSPs, is acyclic, it can be solved in linear time (Freuder 1982). This kind of result has been extended to hypergraphs in (Gottlob, Leone, and Scarcello 2000). Using these theoretical results, some practical methods to solve CSPs have been defined, such as Tree-Clustering (Dechter and Pearl 1989) which can be efficient in practice (Jégou and Terrioux 2003). So, the study of such properties for graphs or hypergraphs has shown its interest regarding the constraint network. Graphs properties have also been exploited to study the properties of compatibility relations for the case of binary CSPs. This is made possible thanks to a representation called microstructure that we can associate to a binary CSP. A microstructure is defined as follows:

Definition 1 (Microstructure) *Given a binary CSP* $P = (X, D, C)$*, the* microstructure *of* P *is the undirected graph* $\mu(P) = (V, E)$ *with:*

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

The transformation of a CSP instance using this representation can be considered as a reduction from the CSP problem to the well known CLIQUE problem (Garey and Johnson 1979) seeing that it can be realized in polynomial time and using the theorem (Jégou 1993) recalled below:

Theorem 1 *An assignment of variables in a binary CSP P is a solution iff this assignment is a clique of size n (the number of variables) in $\mu(P)$.*

The interest to consider the microstructure was firstly shown in (Jégou 1993) in order to detect new tractable classes for CSP based on Graph Theory. Indeed, while determining whether the microstructure contains a clique of size n is NP-complete, this task can be achieved, in some cases, in polynomial time. For example, using a famous result of Gavril (Gavril 1972), Jégou has shown that if the microstructure of a binary CSP is triangulated, then this CSP can be solved in polynomial time. By this way, a new tractable class for binary CSPs has been defined since it is also possible to recognize triangulated graphs in polynomial time. Later, in (Cohen 2003), applying the same approach and also (Gavril 1972), Cohen has shown that the class of binary CSPs with triangulated complement of microstructure is tractable, the achievement of arc-consistency being a decision procedure. More recently, other works have defined new tractable classes of CSPs thanks to the study of microstructure. For example, generalizing the result on triangulated graphs, (Salamon and Jeavons 2008) have shown that the class of binary CSPs the microstructure of which is a *perfect graph* constitutes also a tractable class. Then, in (Cooper, Jeavons, and Salamon 2010), the class BTP, which is defined by forbidden patterns (as for triangulated graphs), has been introduced. After that, (El Mouelhi et al. 2012) also exploit the microstructure, but in another way, by presenting new results on the effectiveness of classical algorithms for solving CSPs when the number of maximal cliques in the microstructure of binary CSPs is bounded by a polynomial. The study of the microstructure has also shown its interest in other fields. For example, for the problem of counting the number of solutions (Angelsmark and Jonsson 2003), or for the study of symmetries in binary CSPs (Cohen et al. 2006; Mears, de la Banda, and Wallace 2009).

Thus, the microstructure appears as an interesting tool for the study of CSPs, or more precisely, for the theoretical study of CSPs. This notion has been studied and exploited in the limited field of binary CSPs, even if the microstructure for n-ary CSPs has already been considered. Indeed, in (Cohen 2003), the complement of the microstructure of a n-ary CSP is defined as a hypergraph:

Definition 2 (Complement of the Microstructure) *Given a binary CSP $P = (X, D, C)$, the* Complement of the Microstructure *of P is the hypergraph $\overline{\mathcal{M}}(P) = (V, E)$ such that:*

- $V = \{(x_i, v_i) : x_i \in X, v_i \in D_{x_i}\}$,
- $E = E_1 \cup E_2$ *such that*
  - $E_1 = \{\ \{(x_i, v_j), (x_i, v_{j'})\}\ \mid\ x_i \in X$ *and* $j \neq j'\}$
  - $E_2 = \{\{(x_{i_1}, v_{i_1}), \ldots (x_{i_k}, v_{i_k})\}\ \mid\ C_i \in C, S(C_i) = \{x_{i_1}, \ldots, x_{i_k}\}$ *and* $(v_{i_1}, \ldots v_{i_k}) \notin R(C_i)\}$.

One can see that for the case of binary CSPs, this definition is a generalization of the microstructure since the Complement of the Microstructure is then exactly the *complement of the graph* of microstructure. Unfortunately, while it is easily possible to consider the complement of a graph, for hypergraphs this notion is not clearly defined in Hypergraph Theory. For example, should we consider all possible hyperedges of the hypergraph (i.e. all the subsets of V) by associating to each one a universal relation? In this case, the size of representation would be potentially exponential w.r.t. the size of the considered instance of CSP. As a consequence, the notion of microstructure for n-ary CSPs is not explicitly defined in (Cohen 2003), and to our knowledge, this question seems to be considered as open today. Moreover, it turns out that this definition of complement of the microstructure has not really been exploited for n-ary CSPs, even in the paper where it is defined since (Cohen 2003) only exploits it for binary CSPs. More generally, exploiting a definition of a microstructure based on hypergraphs seems to be really more difficult than when it is defined by graphs. Indeed, it is well known that the literature of Graph Theory is really more extended than one of Hypergraph Theory. So, the theoretical results and efficient algorithms to manage them are more numerous, offering a larger number of existing tools which can be operated for graphs rather than for hypergraphs.

So, in this paper, to extend this notion to CSPs with constraints of arbitrary arity, we propose another way than the one introduced in (Cohen 2003). We propose to preserve the graph representation rather than the hypergraph representation. This is possible using known representations of constraint networks by graphs. So, we introduce three possible microstructures, based on the *dual representation* (Dechter and Pearl 1987), on the *hidden variable representation* (Rossi, Petrie, and Dhar 1990) and on the *mixed encoding* (Stergiou and Walsh 1999) of n-ary CSPs. We study the basic properties of such microstructures. We also give some possible tracks to exploit these microstructures for future theoretical developments, focusing particularly on extensions of tractable classes to n-ary CSPs.

The next section introduces different possibilities of microstructures for n-ary CSPs while the third section shows some first results exploiting them. The last section presents a conclusion.

Microstructures for n-ary CSPs

As indicated above, the first evocation of the notion of microstructure to non-binary CSPs was proposed by Cohen in (Cohen 2003) and is based on hypergraphs. In contrast, we will propose several microstructures based on graphs. To do this, we will rely on the conversion of non-binary CSPs to binary CSPs. The well known methods are the dual encoding (also called dual representation), the hidden transformation (also called hidden variable representation) and the mixed encoding (also called combined encoding).

Microstructure based on Dual Representation

The dual encoding appeared in CSPs in (Dechter and Pearl 1987). It is based on the graph representation of hypergraphs called *Line Graphs* which has been introduced in the (Hyper)Graph Theory and which are called *Dual Graphs* for

CSPs. This representation was also used before in the field of Relational Database Theory (Dual Graphs were called *Qual Graphs* in (Bernstein and Goodman 1981)). In this encoding, the constraints of the original problem become the variables (also called *dual variables*). The domain of each new variable is exactly the set of tuples allowed by the original constraint. Then a binary constraint links two dual variables if the original constraints share at least one variable (i.e. the intersection between their scopes is not empty). So, this representation allows to define a binary instance of CSP which is equivalent to the considered n-ary instance. The associated microstructure is then immediately obtained considering the microstructure of this equivalent binary CSP:

Definition 3 (DR-Microstructure) *Given a CSP $P = (X, D, C)$ (not necessarily binary), the* Microstructure based on Dual Representation *of P is the undirected graph $\mu_{DR}(P) = (V, E)$ such that:*

- $V = \{(C_i, t_i) : C_i \in C, t_i \in R(C_i)\}$,
- $E = \{ \{(C_i, t_i), (C_j, t_j)\} \mid i \neq j, t_i[S(C_i) \cap S(C_j)] = t_j[S(C_i) \cap S(C_j)]\}$

where $t[Y]$ denotes the restriction of t to the variables of Y.

Note that this definition has firstly been introduced in (El Mouelhi et al. 2013). As for the microstructure, there is a direct relationship between cliques and solutions of CSPs:

Theorem 2 *A CSP P has a solution iff $\mu_{DR}(P)$ has a clique of size e (the number of constraints).*

Proof: By construction, $\mu_{DR}(P)$ is e-partite, and any clique contains at most one vertex (C_i, t_i) per constraint $C_i \in C$. Hence each e-clique of $\mu_{DR}(P)$ has exactly one vertex (C_i, t_i) per constraint $C_i \in C$. By construction of $\mu_{DR}(P)$, any two vertices (C_i, t_i), (C_j, t_j) joined by an edge (in particular, in some clique) satisfy $t_i[S(C_i) \cap S(C_j)] = t_j[S(C_i) \cap S(C_j)]$. Hence all the tuples t_i in a clique join together, and it follows that the e-cliques of $\mu_{DR}(P)$ correspond exactly to tuples t which are joins of one allowed tuple per constraint, that is, to solutions of P. □

Consider the example below which will be used throughout the paper:

Example 1 *$P = (X, D, C)$ has five variables $X = \{x_1, \ldots, x_5\}$ with domains $D = \{D_{x_1} = \{a, a'\}, D_{x_2} = \{b\}, D_{x_3} = \{c\}, D_{x_4} = \{d, d'\}, D_{x_5} = \{e\}\}$. $C = \{C_1, C_2, C_3, C_4\}$ is a set of four constraints with $S(C_1) = \{x_1, x_2\}$, $S(C_2) = \{x_2, x_3, x_5\}$, $S(C_3) = \{x_3, x_4, x_5\}$ and $S(C_4) = \{x_2, x_5\}$. The relations associated to the previous constraints are given by these tables:*

$R(C_1)$	
x_1	x_2
a	*b*
a'	*b*

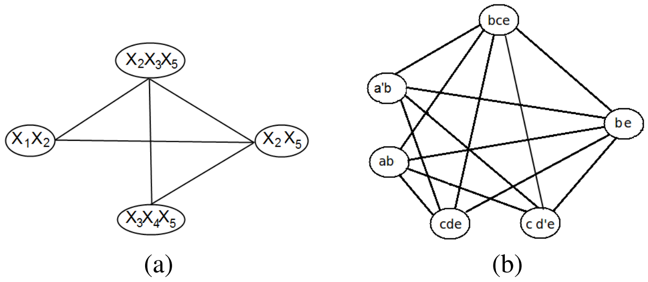
$R(C_2)$		
x_2	x_3	x_5
b	*c*	*e*

$R(C_3)$		
x_3	x_4	x_5
c	*d*	*e*
c	*d'*	*e*

$R(C_4)$	
x_2	x_5
b	*e*

The DR-Microstructure of this example is shown in figure 1. We have 4 constraints, then $e = 4$. Thanks to Theorem 2, a solution of P is a clique of size 4, e.g. $\{ab, bce, be, cde\}$ (in the examples, we denote directly t_i the vertex (C_i, t_i) and v_j the vertex (x_j, v_j) when there is no ambiguity).

Assuming that relations of instances are given by tables (it will be the same for next microstructures), the size of the DR-Microstructure is bounded by a polynomial in the size of the CSP, since $|E| \leq |V|^2$ with $|V| = \Sigma_{C_i \in C} |\{t_i \in R(C_i)\}|$. Moreover, given an instance of CSP, computing its DR-Microstructure can be achieved in polynomial time.



(a) (b)

Figure 1: Dual Graph (a) and DR-Microstructure (b) of the CSP of the example 1.

More generally, with a similar approach, one could define a set of DR-Microstructures for a given n-ary CSP. Indeed, it is known that for some CSPs, some edges of their dual representation can be deleted, while preserving the equivalence (this question has been studied in (Janssen et al. 1989)). In this paper, it is shown that given a hypergraph, we can define a collection of dual (or qual) subgraphs deleting edges while preserving the connexity between shared variables. Some of these subgraphs being minimal for inclusion and also for the number of edges. These graphs can be called *Qual Subgraphs* while the minimal ones are called *Minimal Qual Graphs*. Applying this result, in (Jégou 1991), it is shown that for a given n-ary CSP, there is a collection of equivalent binary CSPs (the maximal one being its dual encoding), assuming that their associated graphs preserve the connexity. So, considering these subgraphs, we can extend the previous definition of DR-Microstructures:

Definition 4 (DSR-Microstructure) *Given a CSP $P = (X, D, C)$ (not necessarily binary) and one of its Dual Subgraph (C, F), the* Microstructure based on Dual Subgraph Representation *of P is the undirected graph $\mu_{DSR}(P, (C, F)) = (V, E)$ with:*

- $V = \{(C_i, t_i) : C_i \in C, t_i \in R(C_i)\}$,
- $E = E_1 \cup E_2$ *such that*
  - $E_1 = \{ \{(C_i, t_i), (C_j, t_j)\} \mid \{(C_i, C_j\} \in F, t_i[S(C_i) \cap S(C_j)] = t_j[S(C_i) \cap S(C_j)]\}$
  - $E_2 = \{ \{(C_i, t_i), (C_j, t_j)\} \mid \{(C_i, C_j\} \notin F\}$.

With this representation, we have the same kind of properties since the size of the DSR-Microstructure is bounded by the same polynomial in the size of the CSP as for DR-Microstructure. Moreover, the computing of the DSR-Microstructure can be achieved in polynomial time. Nevertheless, while Qual Subgraphs are subgraphs of Dual Graph, the DR-Microstructure is a subgraph of the DSR-Microstructure since for each deleted edge, a universal binary relation needs to be considered. Note that the property about the cliques is preserved (the proof of the next theorem is slightly more technical but applies the same approach as the proof of the Theorem 2):

Theorem 3 *A CSP P has a solution iff $\mu_{DSR}(P,(C,F))$ has a clique of size e.*

Microstructure based on Hidden Transformation

The hidden variable encoding was inspired by Peirce (Peirce, Hartshorne, and Weiss 1933) (cited in (Rossi, Petrie, and Dhar 1990)). In the hidden transformation, the set of variables contains the original variables plus the set of dual variables. Then a binary constraint links a dual variable and an original variable if the original variable belongs to the scope of the dual variable. The microstructure is based on this binary representation:

Definition 5 (HT-Microstructure) *Given a CSP $P = (X, D, C)$ (not necessarily binary), the* Microstructure based on Hidden Transformation *of P is the undirected graph $\mu_{HT}(P) = (V, E)$ with:*

- *$V = S_1 \cup S_2$ such that*
  - *$S_1 = \{(x_i, v_i) : x_i \in X, v_i \in D_{x_i}\}$,*
  - *$S_2 = \{(C_i, t_i) : C_i \in C, t_i \in R(C_i)\}$,*
- *$E = \{\ \{(C_i, t_i), (x_j, v_j)\}\ \mid$ either $x_j \in S(C_i)$ and $v_j = t_i[x_j]$ or $x_j \notin S(C_i)\}$.*

Figure 2 represents the HT-Microstructure based on the hidden transformation for the CSP of example 1. We can see that the HT-Microstructure is a bipartite graph. This will affect the representation of solutions. Before that, we should recall that a *biclique* is a complete bipartite subgraph, i.e. a bipartite graph in which every vertex of the first set is connected to all vertices of the second set. A biclique between two subsets of vertices of sizes i and j is denoted $K_{i,j}$. The solutions will correspond to some particular bicliques:

Lemma 1 *In a HT-Microstructure, a $K_{n,e}$ biclique with e tuples, such that no two tuples belong to the same constraint, cannot contain two different values of the same variable.*

Proof: We assume that a $K_{n,e}$ biclique with e tuples, such that no two tuples belong to the same constraint, can contain two different values v_j and v'_j of the same variable x_j. Therefore, there is at least one constraint C_i such that $x_j \in S(C_i)$. Thus, $t_i[x_j]$ = v_j, v'_j or another v''_j. Hence, in all three cases, we have a contradiction since t_i cannot be connected to two different values of the same variable. □

Lemma 2 *In a HT-Microstructure, a $K_{n,e}$ biclique with n values, such that no two values belong to the same variable, cannot contain two different tuples of the same constraint.*

Proof: We assume that a $K_{n,e}$ biclique with n values, such that no two values belong to the same variable, can contain two different tuples t_i and t'_i of the same constraint C_i. Therefore, there is at least one variable x_j such that $t_i[x_j] \neq t'_i[x_j]$. If $v_j = t_i[x_j]$ and $v'_j = t'_i[x_j]$ belong both to the $K_{n,e}$ biclique, we have a contradiction since we cannot have two values of the same variable. □

Using these two lemmas, since a $K_{n,e}$ biclique with n values and e tuples such that no two values belong to the same variable and no two tuples belong to the same constraint corresponds to an assignment on all the variables which satisfies all the constraints, we can deduce the following theorem:

Theorem 4 *Given a CSP $P = (X, D, C)$ and $\mu_{HT}(P)$ its HT-Microstructure, P has a solution iff $\mu_{HT}(P)$ has a $K_{n,e}$ biclique with n values and e tuples such that no two values belong to the same domain and no two tuples belong to the same constraint.*

Based on the previous example, we can easily see that a biclique does not necessarily correspond to a solution. Although $\{a, a', b, c, e, ab, ab', bce, be\}$ is a $K_{5,4}$ biclique, it is not a solution. On the contrary, $\{a, b, c, d, e, ab, bce, be, cde\}$ which is also a $K_{5,4}$ biclique, is a solution of P. Then, the set of solutions is not equivalent to the set of $K_{n,e}$ bicliques, but to the set of $K_{n,e}$ bicliques which contain exactly one vertex per variable and per constraint. This is due to the manner which the graph of microstructure must be completed.

As for DR-Microstructure, the size of the HT-Microstructure is bounded by a polynomial in the size of the CSP, since:

- $|V| = \Sigma_{x_i \in X}|D_{x_i}| + \Sigma_{C_i \in C}|\{t_i \in R(C_i)\}|$ and
- $|E| \leq \Sigma_{x_i \in X}|D_{x_i}| \times \Sigma_{C_i \in C}|\{t_i \in R(C_i)\}|$.

Moreover, given an instance of CSP, computing its HT-Microstructure can also be achieved in polynomial time.

For the third microstructure, we propose another manner to complete the graph of microstructure: this new way of representation is also deduced from hidden encoding.

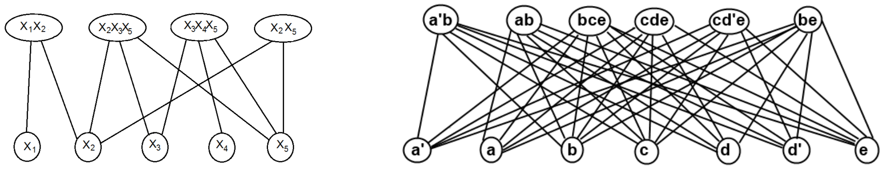
Microstructure based on Mixed Encoding

The Mixed Encoding (Stergiou and Walsh 1999) of n-ary CSPs uses at the same time the dual encoding and the hidden variable encoding. This approach allows us to connect the values of dual variables to the values of original variables, two tuples of two different constraints and two values of two different variables. More precisely:

Definition 6 (ME-Microstructure) *Given a CSP $P = (X, D, C)$ (not necessarily binary), the* Microstructure based on Mixed Encoding *of P is the undirected graph $\mu_{ME}(P) = (V, E)$ with:*

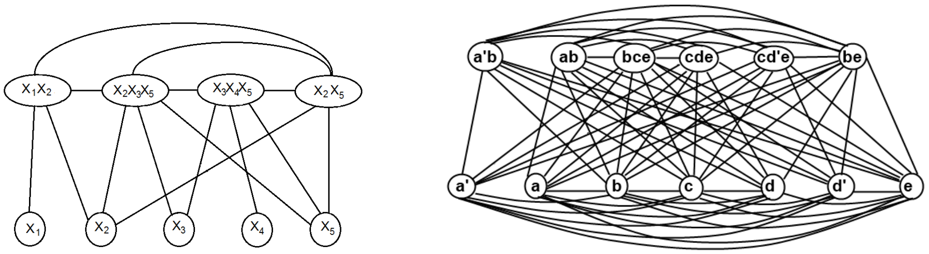
- *$V = S_1 \cup S_2$ such that*
  - *$S_1 = \{(C_i, t_i) : C_i \in C, t_i \in R(C_i)\}$,*
  - *$S_2 = \{(x_j, v_j) : x_j \in X, v_j \in D_{x_j}\}$,*
- *$E = E_1 \cup E_2 \cup E_3$ such that*
  - *$E_1 = \{\ \{(C_i, t_i), (C_j, t_j)\}\ \mid\ i \neq j, t_i[S(C_i) \cap S(C_j)] = t_j[S(C_i) \cap S(C_j)]\}$*
  - *$E_2 = \{\ \{(C_i, t_i), (x_j, v_j)\}\ \mid$ either $x_j \in S(C_i)$ and $v_j = t_i[x_j]$ or $x_j \notin S(C_i)\}$*
  - *$E_3 = \{\ \{(x_i, v_i), (x_j, v_j)\}\ \mid\ x_i \neq x_j\}$.*

The microstructure based on the mixed encoding of the CSP of example 1 is shown in figure 3. We can observe that in this encoding, we have the same set of vertices as for the HT-Microstructure while for edges, we have the edges which belong to the DR-Microstructure and the HT-Microstructure, plus all the edges between values of domains that could appear in the classical microstructure of binary CSPs. This will have an impact on the relationship between the solutions of the CSP and the properties of the



(a) (b)

Figure 2: Hidden graph (a) and HT-Microstructure (b) of the CSP of the example 1.



(a) (b)

Figure 3: Mixed graph (a) and ME-Microstructure (b) of the CSP of the example 1.

graph of ME-Microstructure. The next lemma formalizes these observations:

Lemma 3 *In a ME-Microstructure, a clique on $n + e$ vertices cannot contain two different values of the same variable, neither two different tuples of the same constraint.*

Proof: Let v_i and v'_i be two values of the same variable x_i. By definition, the vertices corresponding to v_i and v'_i cannot be adjacent and so cannot belong to the same clique. Likewise, for the tuples. □

According to this lemma, there is a strong relationship between cliques and solutions of CSPs:

Theorem 5 *A CSP P has a solution iff $\mu_{ME}(P)$ has a clique of size $n + e$.*

Proof: In a ME-Microstructure, according to lemma 3, a clique on $n + e$ vertices contains exactly one vertex per variable and per constraint. So it corresponds to an assignment of n variables which satisfies e constraints, i.e. a solution of P. □

As for other microstructures, the size of the ME-Microstructure is bounded by a polynomial in the size of the CSP, since:

- $|V| = \Sigma_{x_i \in X}|D_{x_i}| + \Sigma_{C_i \in C}|\{t_i \in R(C_i)\}|$ and
- $|E| \leq \Sigma_{x_i \in X}|D_{x_i}| \times \Sigma_{C_i \in C}|\{t_i \in R(C_i)\}| + (\Sigma_{x_i \in X}|D_{x_i}|)^2 + (\Sigma_{C_i \in C}|\{t_i \in R(C_i)\}|)^2$.

Moreover, given an instance of CSP, computing its ME-Microstructure can also be achieved in polynomial time.

Comparisons between microstructures

Firstly, we must observe that none of these microstructures can be considered as a generalization of the classical microstructure of binary CSPs. Indeed, given a binary CSP P, we have $\mu(P) \neq \mu_{DR}(P), \mu(P) \neq \mu_{HT}(P)$ and $\mu(P) \neq \mu_{ME}(P)$.

Moreover, while the DR-Microstructure is exactly the binary microstructure of the dual CSP, neither the HT-Microstructure nor the ME-Microstructure correspond to the classical microstructure of the CSP associated to the binary representations coming from the original instance, because of the way to complete these graphs.

Finally, all these microstructures can be computed in polynomial time. Nevertheless, from a practical viewpoint, they seem to be really difficult to compute and to manipulate explicitly. But it is the same for the classical microstructure of binary CSPs. Indeed, this should require having relations given by tables or to compute all the satisfying tuples. And even if this is the case, except for small instances, this would lead generally to build graphs with a too large number of edges. However, this last point is not really a problem because our motivation in this paper concerns the proposal of new tools for the theoretical study of n-ary CSPs. To this end, the following section presents some first results exploiting these microstructures for defining new tractable classes.

Some results deduced from microstructures

We now present some results which can be deduced from the analysis of these microstructures. For this, we will study three tractable classes, including those corresponding to well known properties as ”0-1-all” (Cooper, Cohen, and Jeavons 1994) and BTP (Cooper, Jeavons, and Salamon 2010) for which it is necessary to make a distinctness between the vertices in the graph, and a third one for which the vertices do not have to be distinguished.

Microstructures and number of maximal cliques

In (El Mouelhi et al. 2012), it is shown that if the number of maximal cliques in the microstructure of a binary CSP (denoted $\omega_{\#}(\mu(P))$) is bounded by a polynomial, then classical algorithms like Backtracking (BT), Forward Checking (FC (Haralick and Elliot 1980)) or Real Full Look-ahead (RFL (Nadel 1988)) solve the corresponding CSP in polynomial time. Exactly, the cost is bounded by $O(n^2 d \cdot \omega_{\#}(\mu(P)))$ for BT and FC, and by $O(ned^2 \cdot \omega_{\#}(\mu(P)))$ for RFL. We analyze here if this kind of result can be extended to n-ary CSPs, exploiting the different microstructures.

More recently in (El Mouelhi et al. 2013), these results have been generalized to n-ary CSPs, exploiting the Dual Representation, using the algorithms nBT, nFC and nRFL, which are the n-ary versions of BT, FC and RFL. More precisely, by exploiting a particular ordering for the assignment of variables, it is shown that the complexity is bounded by $O(nea \cdot d^a \cdot \omega_{\#}(\mu_{DR}(P)))$ for nBT, and by $O(nea \cdot r^2 \cdot \omega_{\#}(\mu_{DR}(P)))$ for nFC and nRFL, where a is the maximum arity for constraints and r is the maximum number of tuples per compatibility relations.

Based on the time complexity of these algorithms, and regarding some classes of graphs with number of maximal cliques bounded by a polynomial, it is easy to define new tractable classes. Such classes of graph are, for example, *planar* graphs, *toroidal* graphs, graphs *embeddable in a surface* (Dujmovic et al. 2011) or *CSG* graphs (Chmeiss and Jégou 1997). This result can be summarized by:

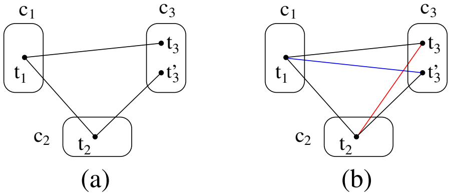
Theorem 6 *CSPs of arbitrary arities whose the DR-Microstructure is either a planar graph, a toroidal graph, a graph embeddable in a surface or a CSG graph, are tractable.*

For HT-Microstructures, such a result does not hold. Indeed, these microstructures are bipartite graphs. So the maximal cliques have size at most two since they correspond to edges and their number is the number of edges in the graph, which is then bounded by a polynomial, independently of the tractability of the instance.

For ME-Microstructures, such a result does not hold too, but for a different reason. By construction, the edges corresponding to the set $E_3 = \{\ \{(x_i, v_i),\ (x_j, v_j)\} \mid\ \ x_i \neq x_j\}$ of definition 6 allow all the possible assignments of variables, making the number of maximal cliques exponential except for CSPs with a single value per domain.

Microstructures and BTP

The property BTP (Broken Triangle Property) (Cooper, Jeavons, and Salamon 2010) defines a new tractable class for binary CSPs while exploiting characteristics of the microstructure. The BTP class turns out to be important because it captures some tractable classes (such as the class of tree-CSPs and other semantical tractable classes such as RRM). The question is then: could we extend this property to n-ary CSPs while exploiting characteristics of their microstructures? A first discussion about this appears in (Cooper, Jeavons, and Salamon 2010). Here, we extend these works, by analyzing the question on the DR, HT and ME-Microstructures. Before, we recall the BTP property:



(a) (b)

Figure 4: DR-Microstructure of a n-ary satisfying BTP on its dual representation

Definition 7 *A CSP instance (X, D, C) satisfies the Broken Triangle Property (BTP) w.r.t. the variable ordering $<$ if, for all triples of variables (x_i, x_j, x_k) s.t. $x_i < x_j < x_k$, s.t. $(v_i, v_j) \in R(C_{ij})$, $(v_i, v_k) \in R(C_{ik})$ and $(v_j, v'_k) \in R(C_{jk})$, then either $(v_i, v'_k) \in R(C_{ik})$ or $(v_j, v_k) \in R(C_{jk})$. If none of these two tuples exist, (v_i, v_j), (v_i, v_k) and (v_j, v'_k) is called a Broken Triangle on x_k.*

In (Cooper, Jeavons, and Salamon 2010), it is shown that, if a binary CSP is BTP, finding a good ordering and solving it is feasible in $O(n^3 d^4 + ed^2)$.

DR-Microstructure. To extend BTP to n-ary CSPs, the authors propose to consider the Dual Graph as a binary CSP, translating directly the BTP property. We denote DBTP this extension. For example, Figure 4 presents the DR-Microstructure of an instance P involving three constraints. In Figure 4 (a), we can observe the presence of a broken triangle on c_3 if we consider the ordering $c_1 \prec c_2 \prec c_3$ and so P does not satisfy DBTP w.r.t. $\prec$. In contrast, in Figure 4(b), if either t_1 and t'_3 (blue edge) or t_2 and t_3 (red edge) are compatible, then P satisfies DBTP according to $\prec$.

But it is possible, analyzing the DR-Microstructure, to extend significantly the first results achieved in (Cooper, Jeavons, and Salamon 2010), these ones being limited to show that the binary tree-structured instances are BTP on their dual representation. For example, it can be shown that for binary CSPs, the properties of the classical microstructure are clearly different than the ones of the associated DR-Microstructure, proving that for a binary instance, the existence of broken triangles is not equivalent, considering one or the other of these two microstructures. Moreover, it can also be proved that if a n-ary CSP has β-acyclic hypergraph (Graham 1979; Fagin 1983), then its DR-Microstructure admits an order such that there is no broken triangle, thus satisfying BTP. Other results due to the properties of the DR-Microstructure can be deduced considering BTP. More details about these results can be found in (El Mouelhi, Jégou, and Terrioux 2013).

HT and ME-Microstructures. For HT-Microstructure, one can easily see that no broken triangle exists explicitly since this graph is bipartite. To analyze BTP on this microstructure, one should need to consider universal constraints (i.e. with universal relations) between vertices of the constraint graph resulting from Hidden Transformation. Also, we will directly study ME-microstructure because this microstructure has the same vertices as the HT-Microstructure and it has been completed with edges between these vertices.

So, consider now the HT-Microstructure. Extending BTP on this microstructure is clearly more complicated because we must consider at least four different cases of triangles, because contrary to BTP or BTP on the DR-Microstructure, we have two kinds of vertices: tuples of relations and values of domains. Moreover, since for BTP, we must also consider orderings such as $i < j < k$, actually we must consider six kinds of triangles since it is possible, for BTP to permute the order of the two first variables: (1) $x_i < x_j < x_k$, (2) $x_i < x_j < C_k$, (3) $x_i < C_j < x_k$ (or $C_i < x_j < x_k$), (4) $x_i < C_j < C_k$ (or $C_i < x_j < C_k$), (5) $C_i < C_j < x_k$, (6) $C_i < C_j < C_k$.

One can notice the existence of a link for BTP between DR-Microstructure and ME-Microstructure. Indeed, if a n-ary CSP P has a broken triangle on DR-Microstructure, for any possible ordering of the constraints, then P possesses necessarily a broken triangle for any ordering on mixed variables (variables and constraints). This leads us to the following theorem which seems to show DR-Microstructure as the most promising one w.r.t. the BTP property:

Theorem 7 *If a CSP P satisfies BTP considering its ME-Microstructure, that is an ordering on mixed variables, then there exists an ordering for which P satisfies BTP considering its DR-Microstructure.*

Microstructures and 0-1-all

In the previous subsections, it seems that the DR-Microstructure should be the most interesting. Does this feeling remains true for other tractable classes? To begin the study we analyze the well known tractable class defined by *Zero-One-All constraints* (”0-1-all”) introduced in (Cooper, Cohen, and Jeavons 1994). Firstly, we recall the definition:

Definition 8 *A binary CSP $P = (X, D, C)$ is said **0-1-all** (ZOA) if for each constraint C_{ij} of C, for each value $v_i \in D_{x_i}$, C_{ij} satisfies one of the following conditions:*

- ***(0)** for any value $v_j \in D_{x_j}, (v_i, v_j) \notin R(C_{ij})$,*
- ***(1)** there is a unique value $v_j \in D_{x_j}$ such that $(v_i, v_j) \in R(C_{ij})$,*
- ***(all)** for any value $v_j \in D_{x_j}, (v_i, v_j) \in R(C_{ij})$.*

This property can be represented graphically using the microstructure. In the case of the DR-Microstructure, it seems easy to define the same kind of property. With respect to the case of the definition given above (the one of (Cooper, Cohen, and Jeavons 1994) defined for binary CSPs), the difference will be related to the fact that the edges of the DR-Microstructure connect now tuples of relations. So, since there is no particular feature which can be immediately deduced from the new representation, the satisfaction of the ”0-1-all” property is obviously related to the properties of the considered instance.

For the HT-Microstructure, now, the edges connect tuples (vertices associated to constraints of the CSP) to values (vertices associated to variables of the CSP). We now analyze these edges from two viewpoints, i.e. from the two possible directions.

- *Edges from the tuples to the values*. Each tuple is connected to the values appearing in the tuple. So, for each constraint associated to the HT-Microstructure, the connection is a ”one” connection, satisfying the conditions of the ”0-1-all” property.
- *Edges coming from the values to the tuples*. For a constraint associated to the HT-Microstructure, a value is connected to the tuples where it appears. We discuss the three possibilities:
  - ”0” connection. A value is supported by no tuple. If we consider a binary CSP, it is the same case as for the classical definition, with a connection ”0”.
  - ”1” connection. A value is supported by one tuple. If we consider a binary CSP, it is also the same case as for the classical definition, with a connection ”1”.
  - ”all” connection. A value is supported by all the tuples of a constraint. We have also the same configuration as for the ”all” connections in the case of binary CSPs.

So, we can deduce the next theorem, which allows us to think that for the HT-Microstructure, we have a representation at least as powerful as for the case of classical microstructure.

Theorem 8 *If a binary CSP P satisfies the ”0-1-all” property, then P satisfies the ”0-1-all” property considering its HT-Microstructure.*

Finally, for the ME-Microstructure, we must verify simultaneously the conditions defined for the DR and HT-Microstructures because, the additional edges connecting vertices associated to values correspond to universal constraints, which trivially satisfy the ”0-1-all” property.

To conclude, by construction, nothing is opposite to satisfy the conditions of ZOA, even if, as for the case of binary CSPs, these conditions are really restrictive.

Conclusion

In this paper, we have introduced the concept of microstructure in the case of CSP with constraints of arbitrary arity. If the concept of microstructure of binary CSP is now well established and has enabled to provide the basis for many theoretical works in CSPs, for the general case, the notion of microstructure was not clearly established before. Also, in this paper, we have wanted to define explicitly a microstructure of CSP for the general case. The idea is to provide a tool for the theoretical study of CSP with constraints of any arity.

It would be possible to define such microstructures in terms of hypergraphs, as suggested in (Cohen 2003). However, given the wealth of literature on the graphs, both for combinatorics and for algorithms, we have preferred to define these microstructures in terms of graphs. Three proposals are presented here: the DR-Microstructure, the HT-Microstructure and the ME-microstructure. Actually, they are derived from the representation of n-ary CSPs by equivalent binary CSPs: the dual representation, the hidden variable transformation, and the mixed approach. We have studied these different microstructures whose none constitutes a formal generalization of the classical binary microstructure. Indeed, if the considered CSP is a binary CSP, none of these microstructures is the conventional binary microstructure.

Although this work is prospective, we have begun to show the interest of this approach. For this, we have studied some known tractable classes which have been initially defined for binary CSPs, and expressed in terms of properties of the microstructure of binary CSPs. Here, a first result is related to the case of microstructures of binary CSP whose the number of maximal cliques is bounded by a polynomial. These instances are known to be tractable in polynomial time by the usual algorithms for solving binary CSPs, as BT, FC or RFL. These classes extend naturally to n-ary CSPs whose microstructures satisfy the same properties about the number of maximal cliques, if now using the n-ary versions of the same algorithms. We have also shown how the BTP class can naturally be extended to non-binary CSPs while expressing the notion of broken triangle within a microstructure of n-ary CSPs. This class is of interest because it includes various well-known tractable classes of binary CSPs, which are now defined in terms of constraints of arbitrary arity.

We now hope that these tools will be used at the level of n-ary CSPs for theoretical studies as it was the case for the classical microstructure of binary CSPs. Although a practical use of these microstructures seems quite difficult for us with respect to issues of efficiency, we believe that one possible and promising track of this work could be to better understand how common backtracking algorithms work efficiently for the n-ary case, and the same thing for numerous heuristics.

Acknowledgments

This work was supported by the French National Research Agency under grant TUPLES (ANR-2010-BLAN-0210).

References

Angelsmark, O., and Jonsson, P. 2003. Improved algorithms for counting solutions in constraint satisfaction problems. In *CP*, 81–95.

Bernstein, P., and Goodman, N. 1981. The power of natural semijoins. *SIAM J. Comput.* 10-4:751–771.

Chmeiss, A., and Jégou, P. 1997. A generalization of chordal graphs and the maximum clique problem. *Information Processing Letters* 62:111–120.

Cohen, D. A.; Jeavons, P.; Jefferson, C.; Petrie, K. E.; and Smith, B. M. 2006. Symmetry definitions for constraint satisfaction problems. *Constraints* 11(2-3):115–137.

Cohen, D. A. 2003. A New Classs of Binary CSPs for which Arc-Constistency Is a Decision Procedure. In *CP*, volume 2833 of *LNCS*. Springer. 807–811.

Cooper, M.; Cohen, D.; and Jeavons, P. 1994. Characterising Tractable Constraints. *Artificial Intelligence* 65(2):347–361.

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

Dechter, R., and Pearl, J. 1987. Network-based heuristics for constraint satisfaction problems. *Artificial Intelligence* 34:1–38.

Dechter, R., and Pearl, J. 1989. Tree-Clustering for Constraint Networks. *Artificial Intelligence* 38:353–366.

Dujmovic, V.; Fijavz, G.; Joret, G.; Sulanke, T.; and Wood, D. R. 2011. On the maximum number of cliques in a graph embedded in a surface. *European J. Combinatorics* 32(8):1244–1252.

El Mouelhi, A.; Jégou, P.; Terrioux, C.; and Zanuttini, B. 2012. On the efficiency of backtracking algorithms for binary constraint satisfaction problems. In *ISAIM*.

El Mouelhi, A.; Jégou, P.; Terrioux, C.; and Zanuttini, B. 2013. Some new tractable classes of csps and their relations with backtracking algorithms. In *CPAIOR*, 61–76.

El Mouelhi, A.; Jégou, P.; and Terrioux, C. 2013. Sur une classe polynomiale hybride pour les csp d'arité quelconque. In *JFPC 2013 (in French), to appear.*

Fagin, R. 1983. Degrees of acyclicity for hypergraphs and relational database schemes. *J. ACM* 30(3):514–550.

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

Garey, M., and Johnson, D. 1979. *Computer and Intractability*. Freeman.

Gavril, F. 1972. Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing* 1 (2):180–187.

Gottlob, G.; Leone, N.; and Scarcello, F. 2000. A Comparison of Structural CSP Decomposition Methods. *Artificial Intelligence* 124:343–282.

Graham, M. H. 1979. On the universal relation. Technical report, University of Toronto.

Haralick, R., and Elliot, G. 1980. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence* 14:263–313.

Janssen, P.; Jégou, P.; Nouguier, B.; and Vilarem, M. 1989. A filtering process for general constraint satisfaction problems: achieving pairwise-consistency using an associated binary representation. In *Proceedings of IEEE Workshop on Tools for Artificial Intelligence*, 420–427.

Jégou, P., and Terrioux, C. 2003. Hybrid backtracking bounded by tree-decomposition of constraint networks. *Artificial Intelligence* 146:43–75.

Jégou, P. 1991. *Contribution à l'étude des problèmes de satisfaction de contraintes : Algorithmes de propagation et de résolution – Propagation de contraintes dans les réseau dynamiques*. Ph.D. Dissertation, Université des Sciences et Techniques du Languedoc.

Jégou, P. 1993. Decomposition of Domains Based on the Micro-Structure of Finite Constraint Satisfaction Problems. In *AAAI*, 731–736.

Mears, C.; de la Banda, M. G.; and Wallace, M. 2009. On implementing symmetry detection. *Constraints* 14(4):443–477.

Nadel, B. 1988. *Tree Search and Arc Consistency in Constraint-Satisfaction Algorithms*. In *Search in Artificial Intelligence*. Springer-Verlag. 287–342.

Peirce, C.; Hartshorne, C.; and Weiss, P. 1933. *Collected Papers of Charles Sanders Peirce*, volume vol. 3. Harvard University Press.

Rossi, F.; Petrie, C. J.; and Dhar, V. 1990. On the equivalence of constraint satisfaction problems. In *ECAI*, 550–556.

Rossi, F.; van Beek, P.; and Walsh, T. 2006. *Handbook of Constraint Programming*. Elsevier.

Salamon, A., and Jeavons, P. 2008. Perfect Constraints Are Tractable. In *CP*, 524–528.

Stergiou, K., and Walsh, T. 1999. Encodings of Non-Binary Constraint Satisfaction Problems. In *AAAI*, 163–168.