Decomposition of domains based on the micro-structure of Finite Constraint-Satisfaction Problems

Philippe Jégou *

L.I.U.P. - Université de Provence
3, place Victor Hugo
F13331 Marseille cedex 3, France
jegou@gyptis.univ-mrs.fr

Abstract

In this paper, we present a method for improving search efficiency in the area of Constraint-Satisfaction-Problems in finite domains. This method is based on the analysis of the "micro-structure" of a CSP. We call micro-structure of a CSP, the graph defined by the compatible relations between variable-value pairs: vertices are these pairs, and edges are defined by pairs of compatible vertices. Given the micro-structure of a CSP, we can realize a pre-processing to simplify the problem with a decomposition of the domains of variables. So, we propose a new approach to problem decomposition in the field of CSPs, well adjusted in cases such as classical decomposition methods are without interest (i.e. when the constraint graph is complete). The method is described in the paper and a complexity analysis is presented, given theoretical justifications of the approach. Furthermore, two polynomial classes of CSPs are induced by this approach, the recognition of them being linear in the size of the instance of CSP considered.

Introduction

Constraint-satisfaction problems (CSPs) involve the assignment of values to variables which are subject to a set of constraints. Examples of CSPs are map coloring, conjunctive queries in a relational databases, line drawings understanding, pattern matching in production rules systems, combinatorial puzzles...

In the general case, finding a solution or testing if a CSP admits a solution is a NP-complete problem. A well known method for solving CSP is the Backtrack procedure. If n is the number of variables, d the size of the domains of variables, and m the number of constraints, the complexity of this procedure is $O(m.d^n)$. A better bound is given using decomposition methods as tree-clustering (Dechter & Pearl 1989) or cycle-cutset method (Dechter 1990). The complexity is then of the order of d^K, K being a parameter function of the structure of the CSP (the constraint graph). If the constraint network is a complete graph, then $K = n$. The decomposition methods are based on the structure of the CSP, i.e. the structure of the constraint graph.

In this paper, we present a decomposition method based on the "micro-structure" of the CSP. We call micro-structure of a CSP, the graph defined by the compatible relations between variable-value pairs: vertices are these pairs, and edges are defined by pairs of compatible vertices (compatible values). Given the graph associated to the micro-structure of a CSP, the problem of finding a solution to the CSP is equivalent to the problem of finding a n-clique (a set of vertices that induces a complete subgraph with these n vertices) in the micro-structure. Considering this property, we use triangulation of graphs (Kjærulff 1990) and clustering of values driven by maximal cliques in the micro-structure to decompose the micro-structure associated to the CSP $\mathcal{P}$ to solve. This approach is motivated by the good algorithmic properties of triangulated graph, particularly to find maximal cliques. Every maximal clique induces a domains decomposition, and so, generates a collection of problems $\mathcal{P}_1$, $\mathcal{P}_2$,... $\mathcal{P}_p$, equivalent to the initial problem $\mathcal{P}$. Each problem $\mathcal{P}_i$, corresponds to a sub-problem of $\mathcal{P}$ with a size of domains equal to δ_i, with the inequality $\delta_i \leq d$. So the complexity of solving $\mathcal{P}$, is now the sum of the complexities $O(m.\delta_i^n)$, for $i = 1, 2,... p$. The complexity of the decomposition is linear in the size of the problem $\mathcal{P}$, and the number of new sub-problems is at most linear in the size of $\mathcal{P}$. The quality of the decomposition is related to the value of each δ_i: more the value δ_i is small, more the decomposition is good. For example, if $\delta_i = 1$ or 2, the complexity of the problem $\mathcal{P}_i$ is now polynomial.

The second section introduces some preliminaries about CSPs while the third section defines formally the micro-structure. The method of domains decomposition is presented in the next section. This is followed by a theoretical analysis of the method, concerning a complexity analysis, and showing some polynomial classes of problems associated to the method.

* This work is supported by the BAHIA project of the PRC-GDR IA of CNRS.

Preliminaries

A finite CSP (Constraint Satisfaction Problem) is defined as a set X of n variables $X_1, X_2, \ldots X_n$, a set D of finite domains $D_1, D_2, \ldots D_n$, and a set C of m constraints $C_1, C_2, \ldots C_m$. A constraint C_i is defined on a set of variables $(X_{i1}, \ldots X_{ij_i})$ by a subset of the cartesian product $D_{i1} \times \ldots D_{ij_i}$; we note this subset R_i (R_i specifies which values of the variables are compatible with each other). R is the set of all Ri, for $i=1 \ldots m$. So, we denote a CSP $\mathcal{P} = (X,D,C,R)$. A solution is an assignment of value to all variables which satisfies all the constraints. For a CSP $\mathcal{P}$, the hypergraph (X,C) is called the constraint hypergraph. A binary CSP is one in which all the constraints are binary, i.e. they involve only pairs of variables, so (X,C) is then a graph (called constraint graph) associated to (X,D,C,R). This paper deals only with binary CSPs. To simplify notations for binary CSPs, a constraint between variables X_i and X_j is denoted C_{ij}, and the associated relation R_{ij}. For a given CSP, the problem is either to find all solutions or one solution, or to know if there exists any solution; the last problem is known to be NP-complete.

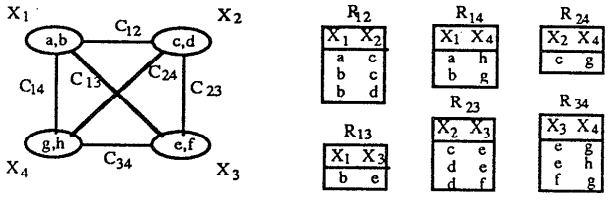

Figure 1. Binary CSP with complete constraint graph.

CSPs are normally solved by different versions of backtrack search. In this case, if d is the size of domains (maximum number of values in domains D_i), the theoretical time complexity of search is then $O(m.d^n)$. Consequently, many works try to improve the search efficiency. They mainly deal with binary CSPs. In (Freuder 1982), Freuder, considering the problem of finding one solution, gives a preprocessing procedure for selecting a good variable ordering prior to running the search. One of his main results is a sufficient condition for backtrack-free search. This condition concerns on one hand a structural property of the constraint graph, and on the other hand a local consistencies. After (Freuder 1982), Dechter and Pearl (Dechter and Pearl 1988) give two classes of polynomially solvable CSPs. For example, they define a property: *if a binary CSP is arc-consistent, and if its constraint graph is acyclic, then the CSP admits a solution and there is a backtrack-free search order.* This property holds for n-ary CSPs with hypergraphs (Janssen et al 1989).

Some methods use decomposition techniques based on structural properties of the CSP. These methods exploit the fact that the tractability of CSPs is intimately connected to the topological structure of their underlying constraint graphs. Moreover, these methods give an upper bound to the complexity of the problem, therefore, an upper bound to the search. The above property gives the goal of the transformation: given a CSP, the result must be an other CSP, equivalent to the first one, whose the structure is a tree. Two methods are based on this principle: the cycle-cutset method (Decther 1990) and tree-clustering scheme (Decther & Pearl 1989).

The cycle-cutset method (CCM) is based on the notion of cycle-cutset. The cycle-cutset of a graph, is a set of vertices such as the deletion of these vertices induces an acyclic graph. CCM is based on the fact that variables assignments changes the effective connectivity of the constraint graph. So, as soon as all the variables of the cycle-cutset are assigned, all the cycles of the constraint graph are cut. Therefore, the resulting problem is tree-structured and Freuder's theorem (Freuder 1982) can be applied to solve it. A property summarizes the method: *if all the variables belonging to the cycle-cutset are instanciated, and if the resulting CSP is arc-consistent, then the problem admits solutions and a backtrack-free order.* So, searching a solution, we can consider that the size of cycle-cutset corresponds to the height of the backtracking. More precisely, if K is the size of the cycle-cutset, the complexity of CCM is $O(m.d^{K+2})$.

Tree-clustering (TC) consists in forming clusters of variables such as the interactions between the clusters is tree structured. The hyper-edges of the induced constraint hypergraph are defined by the clusters of variables. The new CSP is equivalent to the first one, but the associated constraint hypergraph is acyclic. So, the property concerning acyclic n-ary CSPs holds for this CSP. If E is the size of the maximal cluster, the complexity of TC is then $O(n.E.d^E)$.

If the constraint network is a complete graph, we have the equality $E = K+2 = n$. So, the complexity of decomposition methods is the same than for classical backtracking, of the order of d^n. Consequently, complete constraint graphs (n-cliques) can be considered as hard instances of CSP for decomposition methods. The decomposition method described in this paper proposes a solution to handle these hard CSPs, but can also be used on incomplete constraints graph. It is based on a decomposition of the micro-structure of a CSP.

Microstructure of CSPs

We call *micro-structure* of a CSP, the graph defined by the compatible relations between variable-value pairs: vertices are these pairs, and edges are defined by pairs of compatible vertices.

Definition 1. Given a binary CSP $\mathcal{P} = (X,D,C,R)$ such as (X,C) is a complete graph, $\mu(\mathcal{P})$ is called *micro-structure* of $\mathcal{P}$ and it is a n-partite graph defined by

- $X_D = \{ (X_i,a) / X_i \in X \text{ and } a \in D_i \}$
- $C_R = \{ \{(X_i,a),(X_j,b)\} / (X_i,X_j) = C_{ij} \in C \text{ and } (a,b) \in R_{ij} \}$
- $\mu(\mathcal{P}) = (X_D, C_R)$

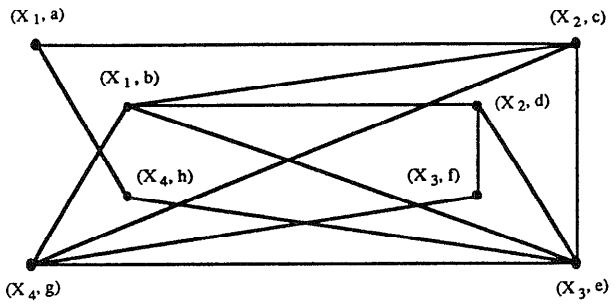

Figure 2. Micro-structure of the CSP given in figure 1.

Necessary, $\mu(\mathcal{P})$ is a n-partite graph because it can not exist edges between vertices of a same domain. In the example in figure 2, we have sets $\{(X_1,a),(X_1,b)\}$, $\{(X_2,c),(X_2,d)\}$, $\{(X_3,e),(X_3,f)\}$ and $\{(X_4,g),(X_4,h)\}$ with no one edge between vertices associated to the same variable, i.e. a set $\{(X_i,\alpha),(X_i,\beta),\ldots\}$.

If (X,C) is not a complete graph, i.e. there are two variables X_i and X_j such as the constraint C_{ij} does not exist between variables X_i and X_j, $\mu(\mathcal{P})$ can be completed adding the universal relation between these variables. The universal relation is the relation $R_{ij} = D_i \times D_j$ (all pairs of values are compatible). In this paper we always consider CSPs with complete constraint graph.

Given a CSP $\mathcal{P} = (X,D,C,R)$ and its micro-structure $\mu(\mathcal{P})$, we can derive a basic property.

Property 2. Given a CSP $\mathcal{P} = (X,D,C,R)$ and its micro-structure $\mu(\mathcal{P})$ we have:

$$(a_1,a_2,\ldots a_n) \text{ is a solution of } \mathcal{P}$$
$$\Leftrightarrow$$
$$\{(X_1,a_1),(X_2,a_2),\ldots (X_n,a_n)\} \text{ is a n-clique of } \mu(\mathcal{P})$$

Proof: $(a_1,a_2,\ldots a_n)$ *is a solution of* $\mathcal{P}$
$\Leftrightarrow \forall i, j, 1 \le i < j \le n, (a_i,a_j) \in R_{ij}$
$\Leftrightarrow \forall i, j, 1 \le i < j \le n, \{(X_i,a_i),(X_j,a_j)\} \in C_R$
$\Leftrightarrow \{(X_1,a_1),\ldots (X_n,a_n)\}$ *is a n-clique of* $\mu(\mathcal{P})$ □

We remark that a solution of $\mathcal{P}$ corresponds to a covering of n vertices in the constraint graph (X,C): there is exactly one vertex (X_i,a) for each domain D_i, for $i = 1, 2,\ldots n$. So, solving a CSP can be considered as the problem of finding a n-clique in its microstructure. The method we present for the decomposition of domains is based on the topological analysis of the microstructure, related to the existence of n-cliques.

Solving CSPs by domains decomposition

We seen that solving a CSP (finding one solution) can be considered as the problem of finding a n-clique in its microstructure. This problem is known to be NP-hard (Karp 1972), but there are classes of graphs such as polynomial (linear) algorithms have been defined. The method we present is based on one of these classes: *triangulated graphs*. So, some definitions and properties must be recalled.

Definition 3. A graph is *triangulated* iff every cycle of length at least four has a chord, i.e. an edge joining two non-consecutive vertices along the cycle.

Property 4. (Fulkerson & Gross 1965) A triangulated graph on n vertices has at most n maximal cliques (a clique is maximal iff it is not included in an other clique).

Property 5. (Gavril 1972) The problem of finding all maximal cliques in a triangulated graph (X,C) is in $O(n+m)$ if $n = /X/$ and $m = /C/$.

Given the micro-structure of any CSP, it is not possible to immediately use these properties because any micro-structure is not necessary a triangulated graph (eg. the micro-structure in the figure 2).

Nevertheless, it is possible to use these results: given any graph $G = (X,C)$, it is possible to add new edges in C to obtain C', such as the graph $\Gamma(G) = (X,C')$ is a triangulated graph. This addition of edges is called triangulation, and can be realized in a linear time in the size of the graph (Kjærulff 1990).

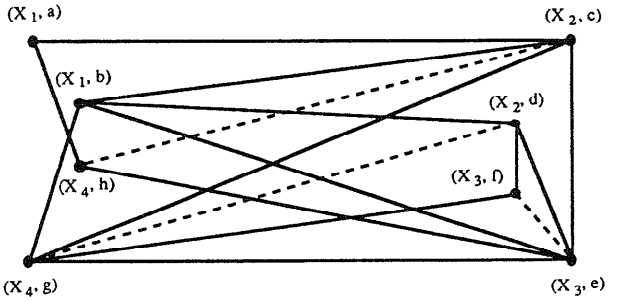

Figure 3. Triangulation of the micro-structure of figure 2. Added edges are given by the dotted lines.

After a triangulation, it is possible to apply the property 5. We show how this approach can be used here.

Suppose we have a CSP $\mathcal{P} = (X,D,C,R)$ and its micro-structure $\mu(\mathcal{P}) = (X_D,C_R)$. Consider a triangulated graph defined by a triangulation of (X_D,C_R). There are three classes of edges in $\Gamma(X_D,C_R)$:

- edges $\{(X_i,a),(X_j,b)\}$ already in $\mu(\mathcal{P})$, i.e. (a,b) is in R_{ij}
- edges $\{(X_i,a),(X_j,b)\}$ / $i \neq j$: adding this edge corresponds to add the tuple (a,b) in R_{ij}
- edges $\{(X_i,a),(X_i,b)\}$: adding this edge has no semantic

Since $\Gamma(X_D,C_R)$. is a triangulated graph, we know that there are no more than n.d maximal cliques in this graph (by property 4), and that it is possible to find them with a linear algorithm (by property 5). Furthermore, we know

that if there exist solutions, anyone is in a maximal clique of this triangulated graph, and consequently, the search of solutions of $\mathcal{P}$ will be limited to the search of solutions on separated problems, each one associated to a maximal clique.

Consider Y, a maximal clique in the triangulated graph $\Gamma(X_D,C_R)$; two possibilities must be considered:

- Y is not a covering of all domains: there is at least one variable X_i of X that does not appear in the vertices (X_i,a) of Y. Consequently, the clique Y does not contain a n-clique that is a covering of all domains, and so there is no solution in Y.

- Y is a covering of all domains. Given Y, we can induce a new CSP, by the projection of vertices in Y on each domains. So, we obtain a collection of domains $D_{Y,i}$ such as $D_{Y,i} \subseteq D_i$, each new domain $D_{Y,i}$ being induced by the vertices (X_i,a) in Y. The constraints of the new CSP associated to Y are the old constraints, restricted to the values in new domains. Searching a solution can be realized on this new CSP. Nevertheless, the fact that Y is a covering of all domains does not guarantee that there is a solution, because the triangulation adds some new edges that connect vertices corresponding to incompatible values.

Theoretical foundations of the method are given below.

Definition 6. Given a binary CSP $\mathcal{P}=(X,D,C,R)$, its micro-structure $\mu(\mathcal{P})=(X_D,C_R)$, and Y a subset of X_D. The CSP induced by Y on $\mathcal{P}$, denoted $\mathcal{P}(Y)$ is defined by:
- $D_Y = \{D_{Y,1},...D_{Y,n}\}$ such as $D_{Y,i} = \{a \in D_i \ / (X_i,a) \in Y\}$
- $R_{Y,ij} = \{ (a,b) \in R_{ij} \ / \ (X_i,a), (X_j,b) \in Y \}$
- $\mathcal{P}(Y) = (X,D_Y,C,R_Y)$

The theorem below define the principle driving the domain decomposition:

Theorem 7. Given a binary CSP $\mathcal{P} = (X,D,C,R)$, its micro-structure $\mu(\mathcal{P})$ and $\mathcal{Y} = \{Y_1,... Y_k\}$, the set of the maximal cliques of $\Gamma(\mu(\mathcal{P}))$, we have:

$$Solutions(\mathcal{P}) = \bigcup_{1 \le i \le k} Solutions(\mathcal{P}(Y_i))$$

Proof:
- *With property 2, we know that any solution of the problem $\mathcal{P}$ is associated to a n-clique. So, this n-clique is necessary included in one set Y_i because in a graph, each clique belongs necessary to a maximal clique of the considered graph. Consequently, the considered solution of $\mathcal{P}$ is necessary a solution of $\mathcal{P}(Y_i)$.*
- *Every solution of a problem $\mathcal{P}(Y_i)$ is a clique in $\mu(\mathcal{P})$ because all the edges of this clique are edges induced by compatible values in $\mathcal{P}$. Consequently, every solution of $\mathcal{P}(Y_i)$ is a solution of $\mathcal{P}$.* □

We remark that a solution of $\mathcal{P}(Y_i)$ can appear as a solution of an other $\mathcal{P}(Y_i)$. In the figure 4, we present the applying of theorem 7 to the example.

Maximal cliques

$$Y_1 = \{(X_1,a),(X_2,c),(X_4,h)\}$$
$$Y_2 = \{(X_1,b),(X_2,c),(X_3,e),(X_4,g)\}$$
$$Y_3 = \{(X_1,b),(X_2,d),(X_3,e),(X_3,f)\}$$
$$Y_4 = \{(X_2,c),(X_3,e),(X_4,h)\}$$
$$Y_5 = \{(X_1,b),(X_3,e),(X_3,f),(X_4,g)\}$$

Decomposed domains

$$D_{Y_1,1} = \{a\}, D_{Y_1,2} = \{c\}, D_{Y_1,3} = \varnothing, D_{Y_1,4} = \{h\}$$
$$D_{Y_2,1} = \{b\}, D_{Y_2,2} = \{c\}, D_{Y_2,3} = \{e\}, D_{Y_2,4} = \{g\}$$
$$D_{Y_3,1} = \{b\}, D_{Y_3,2} = \{d\}, D_{Y_3,3} = \{e,f\}, D_{Y_3,4} = \varnothing$$
$$D_{Y_4,1} = \varnothing, D_{Y_4,2} = \{c\}, D_{Y_4,3} = \{e\}, D_{Y_4,4} = \{h\}$$
$$D_{Y_5,1} = \{b\}, D_{Y_5,2} = \varnothing, D_{Y_5,3} = \{e,f\}, D_{Y_5,4} = \{g\}$$

Figure 4. Applying theorem 7 to the CSP of figure 1. The cliques Y_1, Y_3, Y_4 and Y_5 do not cover all the domains; so the induced sub-problem are not consistent. On the other hand, the cliques Y_2 induces a consistent sub-problem.

Algorithm:

1 - generation of $\mu(\mathcal{P})$

2 - triangulation of $\mu(\mathcal{P})$; we obtain $\Gamma(\mu(\mathcal{P}))$

3 - research of all maximal cliques in $\Gamma(\mu(\mathcal{P}))$; the result of this step is $\mathcal{Y} = \{Y_1,... Y_k\}$

4 - **for** all Y_i in Y **do**
  if Y_i is a covering of all the domains in D
  then solve $\mathcal{P}(Y_i)$ **else** $\mathcal{P}(Y_i)$ has no solution

The first step is realized first with an enumeration of the values of all the domains to obtain the vertices of $\mu(\mathcal{P})$, and secondly, with an enumeration of all the compatible tuples of relations to obtain the edges of $\mu(\mathcal{P})$. If the problem $\mathcal{P}$ has not a complete constraint graph, it is possible to transform it with the addition of the universal constraint between non-connected variables. The second step can be realized using triangulation algorithms - see (Kjærulff 1990). The maximal cliques can be obtained by the algorithm of Gavril (Gavril 1972)(Golumbic 1980).

The last step is first realized with the generation of the problem $\mathcal{P}(Y_i)$: it is sufficient to define new domains based on the vertices in Y_i. Finally, solving $\mathcal{P}(Y_i)$ is possible with a any classical method such as standard backtracking for example.

Theoretical analysis

Complexity analysis

We first give some notations. Given $\mathcal{P} = (X,D,C,R)$ and its micro-structure $\mu(\mathcal{P}) = (X_D,C_R)$.

- n is the number of variables
- d is the maximal number of values in domains, i.e. $\forall i,\ 1 \leq i \leq n,\ |D_i| \leq d$
- N the number of vertices in $\mu(\mathcal{P})$; $N = \sum_{1 \leq i \leq n} |D_i| \leq n.d$
- m is the number of constraints
- M is the number of edges in $\mu(\mathcal{P})$: $M = \sum_{1 \leq i < j \leq n} |R_{ij}|$

and $M \leq N.(N-1)/2 < n^2.d^2$

- p is the number of maximal cliques in $\Gamma(\mu(\mathcal{P}))$

The cost of step 1 of the algorithm is $O(N+M)$. Nevertheless, if (X,C) is not a complete graph, we have $O(n^2.d^2)$. Triangulation step (step 2) is linear is the size of the resultant graph: $O(N+M')$, if M' is the new set of edges after triangulation. Necessary, $M \leq M' < n^2.d^2$. The cost of finding all maximal cliques in $\Gamma(\mu(\mathcal{P}))$ is also linear: $O(N+M')$. By property 4, we know that the number of maximal cliques p satisfies the inequality $p \leq N$.

For the last step, we first evaluate the cost of solving one problem $\mathcal{P}(Y_j)$; it can be bounded by:

$$O(m.(\prod_{1 \leq i \leq n} |D_{Y_j,i}|))$$

So, the cost of the last step, i.e. the cost of solving all sub-problems $\mathcal{P}(Y_j)$, for $j = 1, 2, \ldots p$, is:

$$O(m.(\sum_{1 \leq j \leq p} (\prod_{1 \leq i \leq n} |D_{Y_j,i}|)))$$

The comparison of this cost with respect to the cost of standard backtracking on the initial problem is necessary. The cost of backtracking on $\mathcal{P}$ is

$$O(m.(\prod_{1 \leq i \leq n} |D_i|))$$

If we consider $d = |D_i|$ and $\delta = |D_{Y_j,i}|$, for $i = 1, 2, \ldots n$ and $j = 1, 2, \ldots p$, the comparison between standard backtracking and domains decomposition is now

$$m.d^n \quad \text{VS} \quad m.p.\delta^n$$

or

$$d^n \quad \text{VS} \quad p.\delta^n$$

We know that p is bounded by $n.d$ (cf. property 4). So we give comparison of exponential terms, d^n and δ^n. Suppose that the decomposition induces a simplification of domains, such as we have for example $d = 2.\delta$. The comparison is now

$$d^n \quad \text{VS} \quad \frac{n.d}{2^n}.d^n$$

because $p.\delta^n = p.(d/2)^n = p.(1/2)^n.d^n \leq [n.d.(1/2)^n].d^n$.

Consequently, the decomposition can be very interesting on the instances of problems such as these kind of hypothesis on d and δ hold, i.e. for the problems such as we have $[n.d.(1/2)^n] \ll 1$.

Two trivial polynomial classes

CSPs with triangulated micro-structures. A first polynomial class induced by the domains decomposition is naturally the class of CSPs such as their micro-structure is already triangulated:

Property 8. Let $\mathcal{P}$ be a CSP, and its micro-structure $\mu(\mathcal{P})$. If $\mu(\mathcal{P})$ is a triangulated graph, then the number of solutions of $\mathcal{P}$ is linear in the size of $\mathcal{P}$ and there is a linear algorithm to find all solutions.

Proof. Applying the algorithm, the first step is linear in the size of $\mathcal{P}$. The second step does not add new edges in the micro-structure $\mu(\mathcal{P})$ because $\mu(\mathcal{P})$ is already triangulated graph. The number of maximal cliques is linear in the size of $\mu(\mathcal{P})$, and in the size of $\mathcal{P}$. Finding all these maximal cliques $Y_1, Y_2, \ldots Y_p$, is linear in the size of $\mathcal{P}$. Finally, all the induced sub-problems $\mathcal{P}(Y_j)$ have at most one value in all the domains $D_{Y_j,i}$, for $j = 1, 2, \ldots p$ and $i = 1, 2, \ldots n$. Consequently, a search for any solution will be linear in the number of variables, that is exactly in $O(m)$. □

The interest of this polynomial class is principally that checking for the adherence to it will be linear in the size of any checked instance.

CSPs such as the triangulation of their micro-structures induces domains of size 1 or 2. We now consider the class of CSPs $\mathcal{P}$ such as the triangulation of their micro-structure $\mu(\mathcal{P})$ connects at most two values belonging to the same domain in every obtained maximal cliques.

Property 9. Let $\mathcal{P}$ be a CSP, and its micro-structure $\mu(\mathcal{P})$. If in $\Gamma(\mu(\mathcal{P}))$ there is at most one new edge $\{(X_i,a),(X_i,b)\}$ per domain D_i in every new maximal

cliques then, there is a polynomial algorithm to solve $\mathcal{P}$ (searching for one solution).

Proof. After applying the algorithm for tiangulation of the micro-structure $\mu(\mathcal{P})$, the size of domains in all the induced sub-problems $\mathcal{P}(Y_j)$ is at most two. Consequently, all induced sub-problems can be solved applying the result given in (Dechter 1992). One corollary of this theorem deals for binary CSPs with bivalent domains, and provides a polynomial method to solve this class of CSPs. □

For the same reasons than for the first polynomial class, the interest of this class is also that checking for the adherence will be linear in the size of any checked instance. Moreover, one can observe that the first class is a subclass of the second: already triangulated graphs are graphs such as their triangulation do not add any edge, and consequently, the size of domains induces by maximal cliques is so necessary equal to 1.

Conclusion

We proposed a new method to reduce domains in constraint satisfaction problems. This method is based on the analysis of the micro-structure of CSP, i.e. the structure of the relations between compatibles values of the domains. Given the micro-structure of a CSP, we present a scheme to decompose domains of variables, forming a set of sub-problems such as they have necessary less values than domains in the initial problem. This decomposition is driven by combinatorial properties of triangulated graphs. The complexity analysis of the method shown the theoretical advantages of the approach. Indeed, given a CSP $\mathcal{P}$, if d is the size of domains of the n variables, and if this problem is defined on m constraints, the complexity of any search like standard backtracking, is $O(m.d^n)$. We shown that the method induces the complexity $O(m.p.\delta^n)$ with p being the number of induced sub-problems — p is necessary linear in the size of the problem $\mathcal{P}$ — and δ is the size of new domains, always satisfying $\delta \leq d$. Furthermore, two polynomial classes of CSPs has been defined, the recognition of their elements being linear in the size of instances. Nevertheless, an experimental analysis must now be realized to see practical interest of the approach.

The decomposition method is at present only defined on binary CSPs. Nevertheless, an extension to n-ary CSPs is possible. A way to realize this extension consists in using primal constraint graph. Suppose we have a n-ary CSP with a constraint C_l between three variables; that is $C_l = \{X_i,X_j,X_k\}$. To generate the microstructure, we consider three binary constraints: C_{ij}, C_{ik} and C_{jk}. The associated relations are $R_{ij} = R_l[(X_i,X_j)]$, $R_{ik} = R_l[(X_i,X_k)]$ and $R_{jk} = R_l[(X_j,X_k)]$. This primal representation is not equivalent to the initial n-ary CSP because the new problem is less constrained. But it is sufficient to realize domains decomposition, since the constraints finally considered to solve the initial CSP will be the initial n-ary constraints, with possibly, smallest domains.

Acknowledgements

I would like to thank Philippe Janssen for the helpful pastaga discussion we had on the method discribed in this paper.

References

Dechter R., Enhancement Schemes for Constraint-satisfaction problems: Backjumping, Learning and Cutset Decomposition, *Artificial Intelligence*, 41 (1990) 273-312.

Dechter R. & Pearl J., Network-based heuristics for constraint-satisfaction problems, *Artificial Intelligence*, 34 (1988) 1-38.

Dechter R. & Pearl J., Tree Clustering for Constraint Networks, *Artificial Intelligence*, 38 (1989) 353-366.

Freuder E.C., A sufficient condition for backtrack-free search, *JACM*, 29-1 (1982) 24-32.

Fulkerson D.R. & Gross O., Incidence matrices and interval graphs, *Pacific J. Math.*, 15 (1965) 835-855.

Gavril F., Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph, *SIAM J. Comput*, 1-2 (1972) 180-187.

Golumbic, Algorithmic Graph Theory and Perfect Graphs, *Academic Press, New-York* (1980).

Janssen P., Jégou P., Nouguier B. & Vilarem M.C., A filtering process for general constraint satisfaction problems: achieving pairwise-consistency using an associated binary representation, *Proc. IEEE Workshop on Tools for Artificial Intelligence*, Fairfax, USA (1989) 420-427.

Karp E.C., Reducibility among combinatorial problems, in Complexity of Computer Computation, Miller & Thatcher Eds., Plenum Press, New-York, (1972), 85-103.

Kjærulff U., Triangulation of Graphs - Algorithms Giving Small Total State Space, *Judex R.R. Aalborg*, Denmark (1990)