Partition-k-AC: An Efficient Filtering Technique Combining Domain Partition and Arc Consistency

Hachemi Bennaceur and Mohamed-Salah Affane

Laboratoire d'informatique de Paris-Nord, Institut Galilée,
Avenue Jean-Baptiste Clément, F-93430 Villetaneuse.
{bennaceu,msa@lipn.univ-paris13.fr

Abstract. The constraint propagation process is a powerful tool for solving constraint satisfaction problems (CSPs). We propose a filtering technique which exploits at best this tool in order to improve the pruning efficiency. This technique, combining domain partition and arc consistency, generalizes and improves the pruning efficiency of the arc consistency, and the singleton arc consistency filtering techniques. The presented empirical results show the gain brought by this technique.

1 Introduction

Constraint Satisfaction Problems (CSPs) involve the assignment of values to variables which are subject to a set of constraints. Filtering techniques are important for solving CSPs. Arc consistency (AC) filtering is widely used in practice because of its simplicity and its low space and time complexities. In the last years some classes of filtering techniques have been proposed in order to improve the pruning efficiency of AC, such as the Singleton Arc Consistency (SAC) [Debruyne and Bessière 1997], the Neighborhood Inverse Consistency (NIC) [Freuder and Elfe 1996], the Restricted Path Consistency (RPC) [Berlandier 1995], the Circuit Consistency (CC) [Bennaceur 1994]. The objective of all these techniques is to improve the search solutions by filtering more values than AC while still keeping the advantages of AC. An experimental comparison between these techniques is given in [Debruyne and Bessière 1997]. We propose, in this paper, a new filtering technique called k-Partition-Arc Consistency (k-Partition-AC) having the same objective as the above techniques and improving the pruning efficiency. Given an arc consistent CSP, our filtering technique divides the variable domains into disjoint sub-domains in order to build a set of arc inconsistent CSPs. The filtering of these CSPs by arc consistency may remove a common set of values from each CSP. We show that these values are inconsistent in the original CSP (see property 2).

We illustrate this technique on the following simple example. Figure 1 shows a CSP P, with five variables and five constraints, represented by its constraint graph. We can easily verify that P is arc consistent. By partitionning the domain D_1 into $D_1^1 = \{v_1, v_2\}$ and $D_1^2 = \{v_3, v_4\}$, P is decomposed into two CSPs P_1^1 and P_1^2 such that P has a solution if and only if P_1^1 or P_1^2 has one. Note that

T. Walsh (Ed.): CP 2001, LNCS 2239, pp. 560–564, 2001.

both P_1^1 and P_1^2 can be filtered by arc consistency. The value v_1 of X_4 is arc inconsistent both in P_1^1 and P_1^2, so using property 2 of section 3, we deduce that this value is also inconsistent in P.

This example shows how our method exploits the advantages of the constraint propagation process. We remark that, both in P_1^1 and P_1^2, the value v_1 is detected arc inconsistent after using the constraint propagation process. k-Partition-AC is based on the main idea in order to increase the pruning efficiency of arc consistency: any common inconsistent value of the CSPs obtained by the domain partition is also inconsistent in the original CSP.

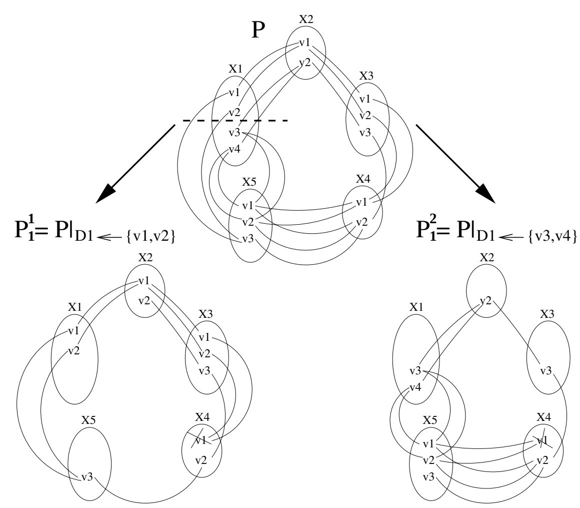

Fig. 1. Example

2 Combining Domain Partition and Arc Consistency

We denote by $P|_{D_i \leftarrow D_i^j}$, $(D_i^j \subset D_i)$, the CSP obtained by restricting the domain D_i to D_i^j in P. Let $P = (\mathcal{X}, \mathcal{D}, \mathcal{C}, \mathcal{R})$ be an arc consistent CSP and X_i be a variable of P. Given a partition $(D_i^1, D_i^2, \ldots, D_i^h)$ of D_i $(h \leq d_i)$, P can be decomposed into h CSPs $P_i^1, P_i^2, \ldots, P_i^h$, $h \leq d_i$, where each CSP P_i^j $1 \leq j \leq h$ is defined by

- $P_i^j = (\mathcal{X}^j, \mathcal{D}^j, \mathcal{C}^j, \mathcal{R}^j)$
- $\mathcal{X}^j \leftarrow \mathcal{X}$
- $\mathcal{C}^j \leftarrow \mathcal{C}$
- $\mathcal{D}^j \leftarrow \{D_r^j,\ 1 \leq r \leq n\}$ where $D_r^j = \begin{cases} D_r \ if\ r \neq i \\ D_i^j \ (D_i^j \subset D_i)\ otherwise. \end{cases}$
- $\mathcal{R}^j \leftarrow \{R_{rs}^j,\ 1 \leq r < s \leq n\}$ with $R_{rs}^j = \begin{cases} R_{rs}\ if\ r \neq i\ and\ s \neq i \\ R_{rs}^j \bigcap (D_r^j \times D_s^j) otherwise. \end{cases}$

Note that: $\bigcup_{j=1}^{h} D_i^j = D_i$ *and* $D_i^{j_1} \bigcap D_i^{j_2} = \emptyset,\ 1 \leq j_1 < j_2 \leq h$. It is clear that the CSPs P_i^1, P_i^2, ..., P_i^h may be arc inconsistent. The following results supply a sufficient condition for a value to be inconsistent in P.

Property 1. Let $v \in D_i^j$, $1 \leq j \leq h$. If v is an arc inconsistent value in P_i^j, then v is an inconsistent value in P.

Property 2. Let $v \in D_r$ $(r \neq i)$. If v is arc inconsistent in each CSP P_i^j, $j = 1, \ldots, h$, then v is an inconsistent value in the original CSP P.

Given a partition P in k_i $(k_i \leq d_i)$ sub-domains $\{D_i^1, D_i^2, \ldots, D_i^{k_i}\}$ of each D_i $(1 \leq i \leq n)$ such that the size of each $\{D_i^j\}$ is bounded by a constant $k \leq d_i$, we have:

Definition 1. *A binary CSP is P-k-arc consistent if and only if* $\forall X_i \in X$, $i = 1, \ldots, n$ *:*

- $D_i \neq \emptyset$*;*
- $\forall D_i^j \subset D_i$, $1 \leq j \leq k_i$, *the CSP* $P_i^j = (P|_{D_i \leftarrow D_i^j})$ *is arc consistent* [1]*. ;*
- $\forall v \in D_r$, $r = 1, \ldots, n$ $r \neq i$, $\exists P_i^j = (P|_{D_i \leftarrow D_i^j})$ *in which the value* v *is arc consistent.*

2.1 Analysis

The technique k-Partition-AC divides a variable domain into disjoint domains, each of them contains at most k elements. The simple way is to divide a domain into singleton sub-domains. Otherwise, many questions can be set here, as what is the size of each sub-domain ? if the size of a sub-domain is known, what are the values that we put in ?. For the moment our aim here is to show that the domain partition combined with a filtering technique as arc consistency can increase the efficiency of these techniques by performing a few more of consistency checks. k-Partition-AC removes a maximum number of values when all the domains are divided into singleton sub-domains. In this particular case, $k = 1$, (all the sub-domains D_i^j are singleton), property 3 shows that 1-Partition-AC removes at least the set of values suppressed by SAC[2].

Property 3. Let P be a binary CSP. If P is 1-arc consistent then P is singleton arc consistent. Conversely if P is singleton arc consistent, P is not necessary 1-arc consistent.

Proof. According to the definition 1, in the particular case, $k = 1$, (all the sub-domains D_i^j are singleton), any 1-arc consistent value is singleton arc consistent. But the reciprocal is not true, see the example in Figure 2 providing a SAC

[1] The definition is here restricted to arc consistency. Naturally, the domain partition may be combined with any filtering concept as the restricted path consistency for instance.

[2] A binary CSP is singleton arc consistent iff $\forall X_i \in X$, $D_i \neq \emptyset$ and $\forall v_r \in D_i$ $P|_{D_i \leftarrow \{v_r\}}$ is arc consistent.

and not 1-Partition-AC CSP. Let us consider the following CSP P where: $X = \{X_1, X_2, X_3, X_4\}$, $D_1 = \{v_1, v_2, v_3\}$, $D_2 = \{v_1, v_2\}$, $D_3 = \{v_1, v_2, v_3\}$ and $D_4 = \{v_1, v_2\}$. The constraints are defined by the relations of Figure 2. Note that there is no constraint between X_2 and X_4:

We can verify that P is singleton arc consistent, since except the value v_1 of D_4, all other values of D_1, D_2, D_3 and D_4 participate in a solution of P. Moreover $P|_{D_4 \leftarrow \{v_1\}}$ is arc consistent. On the contrary, P is not domain arc consistent, since by partitionning the domain D_1 into singleton sub-domains $D_1^1 = \{v_1\}$, $D_1^2 = \{v_2\}$ and $D_1^3 = \{v_3\}$, P is decomposed into three CSPs P_1^1, P_1^2 and P_1^3. The value v_1 of X_4 is arc inconsistent in P_1^1, P_1^2 and P_1^3, then we deduce that this value is 1-arc inconsistent in P.

R_{12}	
X_1	X_2
v_1	v_1
v_2	v_2
v_3	v_1
v_3	v_2

R_{13}	
X_1	X_3
v_1	v_1
v_1	v_3
v_2	v_2
v_2	v_3
v_3	v_1
v_3	v_2
v_3	v_3

R_{14}	
X_1	X_4
v_1	v_1
v_1	v_2
v_2	v_1
v_2	v_2
v_3	v_2

R_{23}	
X_2	X_3
v_1	v_2
v_1	v_3
v_2	v_1
v_2	v_3

R_{34}	
X_3	X_4
v_1	v_1
v_1	v_2
v_2	v_1
v_2	v_2
v_3	v_2

Fig. 2. Table of relations

3 Computational Experiments

This section provides experimental tests on the performances of the k-Partition-AC filtering technique. We have implemented the particular case 1-Partition-AC, using the AC6 algorithm [Bessière 1994], the algorithm k-Partition-AC and detailled experiments are described in [**?**].

The experiments were performed over randomly generated problems using the random model proposed in [Hubbe and Freuder 1992]. We have randomly generated a list of problems according to the following values of the parameters: the number of variables n=100, the domain size d=10, the constraints tightness (the proportion of forbidden value pairs between two constrained variables) $pu = 0.5$ and the graph connectivity (the proportion of the existing constraints) pc varying in $[0, 1]$. The performance measure is the ratio $R = \frac{number\ of\ consistency\ checks}{number\ of\ removed\ values}$. For each problem type (pc fixed) 50 instances were tested. Results reported so far represent the average of the ratio R over the 50 problems for each of the algorithms.

We first compare the behavior of k-Partition-AC relatively to the basic arc consistency filtering AC6 in order to show the impact of k-Partition-AC on arc consistent CSPs (Figure 3). The figures 3 and 4 provide comparing results of our filtering technique and the singleton arc consistency technique.

The experimental results comparing 1-Partition-Arc Consistency with the Arc Consistency and the Singleton Arc Consistency show that the pruning effect

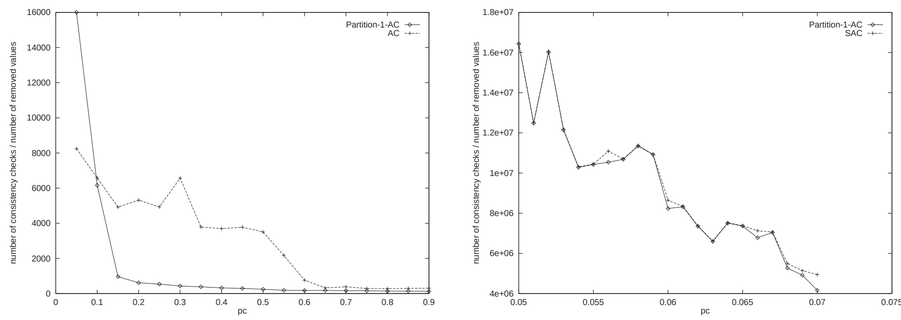

Fig. 3.

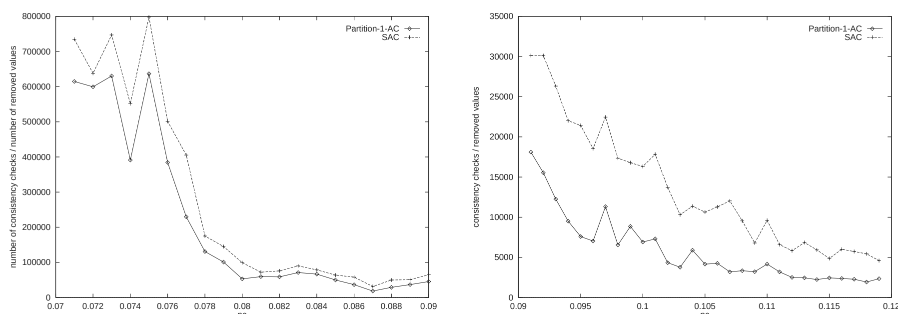

Fig. 4.

is considerably improved and that 1-Partition-AC outperforms SAC on all tested problems.

References

[Bennaceur and Affane 2001] Bennaceur2001 H. Bennaceur and M-S. Affane. Combining Domain Partition and Local Consistencies : An efficient filtering technique. *LIPN report*, 2001.

[Bennaceur 1994] H. Bennaceur. Partial Consistency for Constraint Satisfaction Problems. In *Proc. of the ECAI*, pages 120–125, Amesterdam, Netherlands, 1994.

[Berlandier 1995] P. Berlandier. Improving Domain Filtering Using Restricted Path Consistency. In *Proc. of the IEEE CAIA*, Los Angeles, CA, USA, 1995.

[Bessière 1994] C. Bessière. Arc-consistency and arc-consistency again. *Artificial Intelligence Journal*, 65 (1):179–190, 1994.

[Debruyne and Bessière 1997] R. Debruyne and C. Bessière. Some Practicable Filtering Techniques for the Constraint Satisfaction Problem. In *Proc. of IJCAI*, Nagoya, Japan, 1997.

[Freuder and Elfe 1996] E.C. Freuder and D. Elfe. Neighborhood Inverse Consistency Preprocessing. In *Proc. of the AAAI*, pages 202–208, Portland, OR, USA, 1996.