

# USING FINITE DOMAINS IN CONSTRAINT SATISFACTION PROBLEM

Ilie Popescu

*Computer Science Department , University of Quebec in Outaouais , Gatineau, QuébecCanada, J8X 3X7*

**Abstract:** Constraint satisfaction problem (CSP) methodologies are intended to solve (mostly combinatorial) problems especially in areas of planning and scheduling. Our paper focuses on a model enforcing arc consistency without changing the structure of the constraint network (CN), i.e., only by efficiently removing values from variable domains. The propagation of a variable domain to the constraints related to this variable allows the model to keep under control the size of the search space, by enumerating only the variable values which are part of a solution.

**Key words:** Constraint satisfaction problem, consistency, search space

## 1. INTRODUCTION

Constraint satisfaction problems (CSPs) are very convenient formalisms for many real-life problems, like resource allocation, scheduling and planning problems, database systems, natural language processing, electrical circuit design, and many others [4,5,7]. Many combinatorial problems can be naturally expressed in the frame of CSP.

Finding a solution in a constraint network involves looking for value assignments for a set of variables so that all the constraints are simultaneously satisfied [3,6,8]. The proposed algorithms to solve these kind of problems make a systematic exploration of the search space, changing the original CSP into a new one removing values from domain variables which cannot belong to any solution of the CSP [2]. As long as the unassigned variables have values consistent with the current state of the CSP,

they extend it by assigning values to these variables and change the current state to a new one. This process is called consistency technique and it is the core of most constraint satisfaction methodologies.

In this paper, we present a model based on constraint propagation. The motivation behind this is the desire to reduce the vast amount of deductions required by an expert system when executed on a logic program system. A key idea is to represent the relations among the rules as constraints and to integrate the rule chaining with constraint solving.

In order to reduce the size of the search space, we use the filtering techniques to enforce arc consistency by removing all local inconsistencies that otherwise would have been repeatedly found during search.

## 2. PRELIMINARIES

A CSP is defined as a set of  $n$  variables  $X = (X_1, \dots, X_n)$ , a set of  $n$  domains  $D = \{D_1, \dots, D_n\}$  and a set of binary constraints between pairs of variables  $X_i$  and  $X_j$ ,  $C = \{C_{ij}(X_i, X_j)\}$   $i = 1..n$ ,  $j = 1..n$ ,  $i \neq j$ .

$C_{ij}(v_i, v_j)$  is the Boolean value obtained when variables  $X_i$  and  $X_j$  are replaced by the values  $v_i$  and  $v_j$  respectively.

A solution to a constraint satisfaction problem is a function  $f: X \rightarrow D^n$  such that all the sets of values  $\{v_1, \dots, v_n\} \in D^n = D_1 \times \dots \times D_n$  for the variables  $X_1, \dots, X_n$  satisfy all the constraints belonging to  $R$ .

An arc  $(X_i, X_j)$  is arc consistent with respect to Cartesian product  $D_i \times D_j$  iff  $\forall v_i \in D_i, \exists v_j \in D_j$  such that  $(v_i, v_j) \in C_{ij}$ . Arc consistency is widely used in solving CSPs because it only changes the domains of variables.

## 3. CONSISTENCIES ENFORCEMENT

Consistency is a process of removing values from domains which cannot be part of any solution of a CSP. The basic idea of consistency techniques is that if we can find an inconsistency in the CSP formulation, then tuples of values from the domains of variables implied in a constraint can be removed. By removing inconsistencies, we reduce a CSP to an equivalent but tighter problem. The arc consistency (AC) is the most widely used consistency technique [1]. There exists several arc consistency algorithms starting from AC-1 and concluding at AC-8. These algorithms are based on repeated revisions of variable domains by removing values which violate some constraints, until a consistent graph is reached.

In order to make an arc  $(X_i, X_j)$  consistent, the values  $VX_i$  of the variable  $X_i$  without support in the domain of  $X_j$  are deleted from the domain of  $X_i$ . If a value is not supported, it is removed from its domain. The main drawbacks of these algorithms are those many pairs of values are tested for consistency in every arc revision. These tests are repeated every time an arc is revised.

#### 4. CONSTRAINT OPTIMISATION

In many real-life problems, we do not want any solution but a good solution. Based on the above observation instead of handling a queue of arcs to be propagated, we can use a list of variables that have seen their domain modified. To implement this idea you can carry out a top-down procedure in the following four steps:

- I. Select and represent the variable domains as lists of values
- II. For each variable  $X_i$  and each arc  $C_{ij}(X_i, X_j)$  reduce the domain of  $X_i$ , propagate the new domain of  $X_i$  to all constraints  $C_{ki}(X_k, X_i)$  and delete all the values of  $X_k$  which are not viable.
- III. Add/delete new constraints on the existing variables.
- IV. Stop the propagation when there are not changes in any variable domain or a variable domain is empty.

This model handles constrained problem arising in both static and dynamic environments and provides a general schema for implementing arc consistency techniques. The efficiency of this methodology resides mainly in the fact that the reduced domain of a variable  $X_i$  is propagated to all constraints  $C_{ij}(X_k, X_i)$ ,  $k \neq i$ , implying the variable  $X_i$ : the domains of  $X_k$ ,  $k = 1, 2, \dots$  are reduced by eliminating values which lost their support in the new domain of  $X_i$ . On the other hand, the users of these systems often need to extend the constraints system with some constraints that are specific to the application at hand. This "glass-box" approach extends the well known constraints programming paradigm as a black-box. Applying traditional methods often suffice for developing industrial applications, but understanding their performances requires tools able to reinforce/modify the search strategy by adding/deleting constraints during the search progress.

#### 5. CONCLUSIONS

In this paper we have proposed a new model for solving CSPs that cope with the complexity NP-complete problems. The arc consistency techniques approach is essential to reduce the size of the search space and so to improve

the efficiency of search algorithms. They can be used during a preprocessing filtering step to remove the local inconsistencies that otherwise are repeatedly found during the search.

The solution computation may still be too expensive time wise when the CSP is very large. One of the lessons learned so far from the application of constraint programming tools in practice is that domain specific information is needed. In such cases the model allows the addition/deletion of new domain-specific and problem-specific knowledge. Judging the importance of consistency techniques, integrating new constraints enhance the efficiency of the CSP solver.

## REFERENCES

1. R. Bartak. On-line guide to Constraint Programming, Prague, 1998, <http://kti.mff.cusni.cz/~bartak/constraints/>.
2. R. Debruyne, C. Bessière. Domains Filtering Consistencies. *Journal of Artificial Intelligence Research* 14, pages 205-230, 2001.
3. R. Dechter. Constraint networks. In Stuart C. Shapiro, editor, *Encyclopedia of AI* (second edition), volume 1, pages 276-284. John Wiley & Sons, 1992.
4. A.K. Mackworth. Constraint satisfaction. In Stuart C. Shapiro, editor, *Encyclopedia of AI* (second edition), volume 1, pages 285-293. John Wiley & Sons, 1992.
5. J.J. McGregor. Relational consistency algorithms and their applications in finding sub graph and graph isomorphism, *Information Science* 19, pages 229-250, 1979.
6. P. Meseguer. Constraint satisfaction problems: An overview. *AICOM*, 2, pages 3-17, 1989.
7. U. Montanari. Networks of constraints: Fundamental properties and applications to picture processing, *Information Science* (7) (2), pages 95-132, 1974.
8. E. Tsang. *Foundations of Constraint Satisfaction*. London, Academic Press, 1993.