CSP Search with Responsibility Sets and Kernels

Igor Razgon and Amnon Meisels
Department of Computer Science,
Ben-Gurion University of the Negev,
Beer-Sheva, 84-105, Israel
{irazgon,am}@cs.bgu.ac.il

Abstract

We introduce data structures called responsibility set and kernel. We present an algorithm FC-RK, which is a modification of FC that maintains these structures and uses them for pruning of the search space. According to our experimental evaluation, FC-RK outperforms FC-CBJ on constraint networks encoding graph k-coloring instances and on non-dense random binary constraint networks.

1 Introduction

The present paper introduces a modification of the forward checking algorithm (FC) [Haralick and Elliott, 1980] that associates every removed value with two data structures: the *responsibility set* and the *kernel*. Having introduced the structures, we describe how they help to prune the search space. We define a notion of *filtering value*. Next, we prove the central theorem of the paper which claims that the existence of a filtering value in some state that occurs during work of FC means that the current partial solution of the state is a nogood. We modify FC so that every time after lookahead or backtrack, it checks whether there is a filtering value in the current state. If there is, it backtracks again. We call the resulting modification FC-RK.

The rest of the paper is organized as follows. Section 2 provides the necessary definitions and notations. Section 3 formulates the main theorem and introduces FC-RK. Section 4 briefly discusses the results of preliminary evaluation.

2 Preliminaries and Notations

A binary *constraint network* (CN) $Z = \langle V, D, C\rangle$ is a triple consisting of a set of *variables* V, a set of *domains* D and a set of *constraints* C. Let $V = \{v_1, \ldots, v_n\}$. Then $D = \{D_{v_1}, \ldots, D_{v_n}\}$, where D_{v_i} is the domain of values of variable v_i, $C = \{C_{v_i,v_j} | i \neq j, 1 \leq i,j \leq n\}$, where $c_{v_i,v_j} \subseteq D_{v_i} \times D_{v_j}$ is the set of all compatible pairs of values of v_i and v_j. We refer to the parts of Z as $V(Z)$, $D(Z)$, and $C(Z)$. To emphasize that a value val belongs to the domain of a variable v, we refer to this value as val^v. An *assignment* of a CN Z is a pair $\langle v_i, val\rangle$ such that $v_i \in V(Z)$, $val \in D_{v_i}$. A consistent set of assignments is a *partial solution* of Z. A partial solution that assigns all the variables of Z is a *solution* of Z. A CN that has no solution is *insoluble*. Let P be a partial solution of Z. We denote the set of variables assigned by P by $V(P)$. A *nogood* of a CN Z is a partial solution of Z that cannot be extended to a full solution.

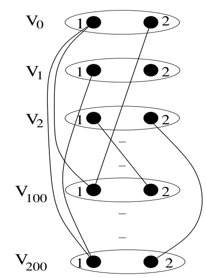

Figure 1: A constraint network.

Throughout the paper we use the CN shown in Figure 1. The ellipses represent the domains of variables and incompatible pairs of values are connected by arcs. For example, value 2 of V_0 is incompatible with value 1 of V_{100}.

3 Responsibility sets, kernels, and FC-RK

In this section we develop a modification of FC called FC-RK. First we define two structures maintained by these algorithms.

Definition 1 *Let P be a nogood of a CN Z. A responsibility set R of P is a subset of $V(Z) \setminus V(P)$ such that there is no consistent extension of P that assigns all the variables of R.*

Definition 2 *Let P be a nogood of a CN Z, let R be a responsibility set of P and let $\langle u, val\rangle \in P$. The kernel of val^u with respect to P and R is the subset of R which contains all the variables whose domains have values incompatible with val^u but compatible with the rest of assignments of P.*

Consider the CN shown on Figure 1. The nogood $\{\langle V_0, 2\rangle\langle V_1, 1\rangle\}$ has a responsibility set $\{V_2, V_{100}, V_{200}\}$, while the kernel of $\langle V_1, 1\rangle$ is $\{V_{200}\}$.

Now we present a modification of FC that maintains responsibility sets and kernels associated with removed values. The proposed method is a slight modification of the technique described in [Schiex and Verfaillie, 1994].

Recall that FC detects that the current partial solution P is a nogood if the current domain of some future variable is emptied or if the current domain of the variable being assigned is wiped out [Prosser, 1993]. In both of these cases, FC discards the last assignment $\langle u, val \rangle$ of the current partial solution and removes val from the current domain of u. We say that P is the nogood *associated* with val^u. The modified version of FC associates val^u with two sets denoted by $rs(val^u)$ and $ker(val^u)$. These are the responsibility set and the kernel of the nogood associated with val^u. The additional operations performed by the modified version of FC are the following.

- The rs and ker sets are initialized to $\emptyset$ for all the values of the CN.
- If a value val_1^v is deleted during the lookahead stage performed after assigning val to u then v is inserted into $ker(val^u)$.
- Consider a value val^u that is deleted during backtrack and let v be the variable whose empty domain caused the backtrack. Then $rs(val^u)$ is set to $S \cup \{v\}$, where S is the union of the rs-sets of all values of v. Also $ker(val^u)$ is updated to the intersection of the current value of $ker(val^u)$ with $rs(val^u)$. (Note that the preliminary value of $ker(val^u)$ is computed at the lookahead stage in order to avoid recomputation of values that are incompatible with val^u.)
- Once a discarded value is restored to the current domain, the rs and ker sets of the value are re-initialized to $\emptyset$.

Let us illustrate the method by simulating a few iterations of the modified FC that processes the CN of Figure 1. It starts by assigning $\langle V_0, 1 \rangle$ $\langle V_1, 1 \rangle$ and then $\langle V_2, 1 \rangle$. The last assignment empties the current domain of V_{100} and FC backtracks. The rs-set of 1^{V_2} is set to $\{V_{100}\}$ so is $ker(1^{V_2})$. Note that the assignment $\langle V_2, 1 \rangle$ can remove values from many other variables of the CN but the kernel is restricted to $\{V_{100}\}$ during the backtrack. In the next step, FC assigns V_2 with 2 and backtracks again because V_{200} is emptied. Then the domain of V_2 is emptied therefore FC backtracks yet another time and produces a nogood $\{\langle V_0, 1 \rangle, \langle V_1, 1 \rangle\}$ with $rs(1^{V_1}) = \{V_2, V_{100}, V_{200}\}$ and $ker(1^{V_1}) = \emptyset$.

Consider a state of FC that occurs just after lookahead or backtrack. Denote the current partial solution by P and its last assignment by $\langle u, val \rangle$. For the given state, define a *filtering value*.

Definition 3 *A value val_1^v removed by backtrack is a filtering value if the following conditions hold.*

- **Compatibility condition.** *The assignments of variables of $V(P) \cap ker(val_1^v)$ are compatible with val_1^v.*
- **Inclusion condition.** *For every unassigned $w \in ker(val_1^v)$, all values of the current domain of w are compatible with val_1^v.*

Now we are ready to formulate the central theorem.

Theorem 1 *Assume that in the given state there is a filtering value val_1^v. Then P is a nogood with a responsibility set $S = rs(val_1^v) \setminus V(P) \cup S^*$, where S^* is the union of rs-sets of all values that belong to the domains of unassigned variables of $ker(val_1^v)$.*

Let us illustrate the theorem on the CN in Figure 1. Consider a state of a CSP solver in which the value 1^{V_0} is removed and associated with the rs-set $\{V_2, V_{100}, V_{200}\}$ and the ker-set $\{V_{100}, V_{200}\}$. Continuing the simulation, we assign V_0 with 2 and V_1 with 1. For the obtained partial solution, 1^{V_0} is a filtering value, all values of the current domains of V_{100} and V_{200} are compatible with 1^{V_0} (the inclusion condition).

Theorem 1 suggests a pruning procedure that discards the current partial solution if it finds a filtering value. The procedure is applied every time after lookahead or backtrack. Algorithm 1 presents the pseudocode of the procedure.

Algorithm 1 PRUNING PROCEDURE OF FC-RK

for every removed value val_1^v **do**
    if val_1^v is a filtering value **then**
        Let $\langle u, val \rangle$ be the last assignment of the current partial solution
        Backtrack with setting $rs(val^u)$ to S (S is defined in Theorem 1)
    end if
end for

We call the modification of FC that computes rs and ker sets and applies Algorithm 1 every time after lookahead and backtrack, FC-RK (RK abbreviates Responsibility sets and Kernels).

4 Preliminary Evaluation

The experimental evaluation shows that, FC-RK outperforms FC-CBJ on the majority of instances of graph k-coloring problem and on binary random CSPs with low density. For dense random CSPs, FC-RK performs more consistency checks than FC-CBJ but visits less nodes on the search tree. We believe that performance of FC-RK could be improved by an algorithm for checking the existence of filtering values, that takes less consistency checks than our current implementation.

References

[Haralick and Elliott, 1980] R. M. Haralick and G.L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14:263–313, 1980.

[Prosser, 1993] P. Prosser. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence*, 9:268–299, 1993.

[Schiex and Verfaillie, 1994] T. Schiex and G. Verfaillie. Nogood Recording for Static and Dynamic Constraint Satisfaction Problem. *International Journal of Artificial Intelligence Tools*, 3(2):187–207, 1994.