A Covering Problem for Hypercubes

Jörg Hoffmann
Max-Planck-Institut für Informatik
Saarbrücken, Germany
hoffmann@mpi-sb.mpg.de

Sebastian Kupferschmid
Universität Freiburg, Institut für Informatik
Freiburg, Germany
kupfersc@informatik.uni-freiburg.de

Abstract

We introduce a new NP-complete problem asking if a "query" hypercube is (not) covered by a set of other "evidence" hypercubes. This comes down to a form of constraint reasoning asking for the satisfiability of a CNF formula where the logical atoms are inequalities over single variables, with possibly infinite variable domains. We empirically investigate the location of the phase transition regions in two random distributions of problem instances. We introduce a solution method that iteratively constructs a representation of the non-covered part of the query cube. In particular, the method is not based on backtracking. Our experiments show that the method is, in a significant range of instances, superior to the backtracking method that results from translation to SAT, and application of a state-of-the-art DP-based SAT solver.

This paper is an extended abstract. More details can be found in the long version of the paper [Hoffmann and Kupferschmid, 2005].

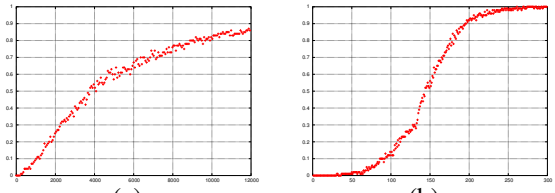
We introduce a new NP-complete problem asking if there is a point in a given n-dimensional "query" hypercube that is not covered by – contained in the union of – a set of other n-dimensional "evidence" hypercubes. An n-dimensional hypercube is a cross product of n intervals. Intervals in our context are defined as statements of the form $l < [\leq]\ x < [\leq]\ u$ where "x" is a variable, "l" and "u" are members of x's domain, and "$<$" is a total order defined over this domain.[1]

Definition 1 *Let QCOVER denote the following problem: Given an n-dimensional hypercube Q, and a set $\mathcal{E}$ of n-dimensional hypercubes, is there a point in Q that is not contained in $\bigcup_{E\in\mathcal{E}} E$?*

Covering problems of this kind arise, e.g., in the context of regression planning with numeric state variables [Koehler, 1998]. More generally, QCOVER is a form of constraint reasoning asking for the satisfiability of a CNF formula where the logical atoms are inequalities over single variables, with possibly infinite variable domains. The correspondence is the following. A QCOVER instance is a constraint problem with n variables x^d. The query hypercube specifies a region inside which the solution must lie, the evidence cubes specify regions inside which the solution must *not* lie. A hypercube corresponds to a conjunction of inequalities of the form $c < [\leq, >, \geq]\ x^d$. So the complement of a hypercube (of an evidence hypercube) corresponds to a *dis*junction of such inequalities, and the overall problem is a conjunction of disjunctive constraints. Vice versa, any conjunction of such disjunctive constraints can be expressed as hypercubes (if a disjunctive constraint does not mention a variable x^d, then the interval in dimension d is the whole variable domain).

Proposition 1 *QCOVER is NP-complete.*

We empirically explore two random distributions of QCOVER instances. The first one, which we call *Random QCOVER*, chooses the end points for all intervals uniformly from a set of m possible values. The second one, which we call *Random 3-QCOVER*, is similar to the fixed clause-length model for generating random 3SAT instances [Mitchell *et al.*, 1992]. It always selects the query cube to be the cross-product of the (whole) variable domains, and, in the evidence cubes, assigns the whole variable domains to all but 3 randomly chosen dimensions. For both distributions, we investigate the location of the phase transition regions. As it turns out, Random 3-QCOVER shows a typical phase transition behaviour while Random QCOVER shows no such behaviour, see Figure 1.



(a) (b)

Figure 1: Typical plot of the proportion of unsatisfiable instances against k for (a) Random QCOVER and (b) Random 3-QCOVER.

By *50% point*, we denote the number k of evidence cubes at which our random instances have (empirically) equal probability of being satisfiable or unsatisfiable. The 50% point depends on n and m. We use instances from the 50% points to evaluate different solution methods.

[1] By square parentheses "$symb1\ [symb2]$" we denote alternative possibilities, i.e. that $symb2$ can be substituted for $symb1$.

Today, in most cases the empirically most efficient solution methods for satisfiability problems are backtracking methods. Such methods are depth-first searches that split, in each search node, the search space along the possible values of a variable. A polynomial propagation of constraints is used to determine conflicts early on in the search tree, and analysis of conflicts is used to prune unnecessary branches. Methods of this kind have proved successful for solving constraint satisfaction problems. In particular, the modern descendants of the Davis Putnam procedure still constitute the state-of-the-art in determining the solvability of propositional CNF formulas. In our work, we have implemented a backtracking method for QCOVER by plugging a straightforward translation to SAT into Chaff [Moskewicz *et al.*, 2001].[2] To contrast this method, we have also developed an algorithm, named *cube elimination*, that, instead of backtracking over possible variable values, iteratively constructs a representation of the non-covered part of the query cube. See Figure 2.

procedure cube elimination(Q, $\mathcal{E}$)
$\mathcal{Q} := \{Q\}$
for all $E \in \mathcal{E}$ **do**
  $\mathcal{Q}' := \emptyset$
  for all $Q' \in \mathcal{Q}$ **do**
    $\mathcal{Q}' := \mathcal{Q}' \cup$ minimal cover($Q' \setminus E$)
  endfor
  $\mathcal{Q} := \mathcal{Q}'$
  if $\mathcal{Q} = \emptyset$ **then** answer "unsatisfiable", **return** $\emptyset$ **endif**
endfor
answer "satisfiable", **return** $\mathcal{Q}$

Figure 2: Cube elimination.

The algorithm maintains a set $\mathcal{Q}$ of hypercubes that initially contains only the query cube itself. Then iteratively all evidence cubes E are "eliminated" by subtracting them from all cubes Q in $\mathcal{Q}$. The result $Q \setminus E$ of such a subtraction is not necessarily a hypercube; we represent it as a *set* of hypercubes, computed by the *minimal cover* procedure. The latter is a simple **for**-loop over all dimensions, returning a set of hypercubes that covers exactly $Q \setminus E$, and that is minimal in the sense that there is no smaller set of hypercubes covering exactly $Q \setminus E$ (the worst-case size of the set is $2n$).

In SAT, where all variables are boolean and have only two possible values, cube elimination comes down to transforming the CNF into a DNF. This seems a hopeless approach, but, for our Random QCOVER distribution, our experiments show that cube elimination is often superior to the backtracking method implemented by Chaff. Figure 3 shows our results for the Random QCOVER distribution, in terms of (nr. of search decisions made by Chaff) divided by (total nr. of cubes generated by cube elimination). Clearly, cube elimination becomes superior as the value of m increases. This is particularly true in the unsatisfiable instances where the cube elimination search space is, in the largest instances, around 5 orders of magnitude smaller than that of backtracking. It should be noted that cube elimination produces a representation of *all* satisfying assignments in the satisfiable cases.

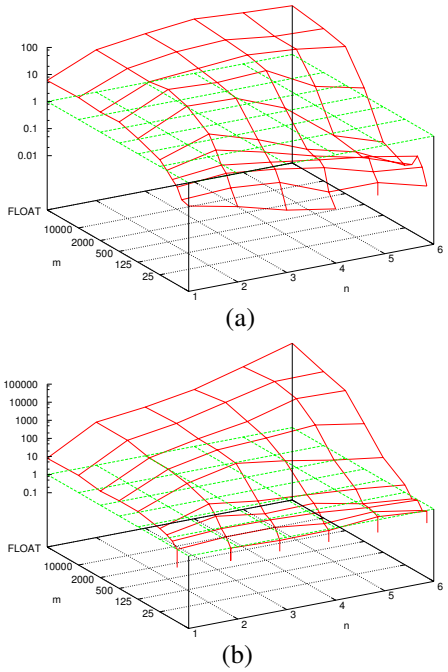


Figure 3: Search space size quotient Chaff vs. cube elimination in Random QCOVER, averaged over (a) all instances, (b) only unsatisfiable instances. The z-axis is log-scaled, the plain $z = 1$ is included for orientation.

In the Random 3-QCOVER distribution, we found backtracking to be generally superior to cube elimination. For lack of space, we ommit the details. In spirit, cube elimination is somewhat similar to the "bucket elimination" framework defined by Rina Dechter [1999]. The cube elimination algorithm is easiest to understand, and was originally motivated by, viewing the satisfiability problem we consider as a *geometrical* problem. This opens up the question if geometrical interpretations of other satisfiability problems can lead to interesting new methods for these problems.

References

[Dechter, 1999] R. Dechter. Bucket elimination: A unifying framework for reasoning. *AI*, 113:41–85, 1999.

[Hoffmann and Kupferschmid, 2005] J. Hoffmann and S. Kupferschmid. A covering problem for hypercubes. TR 218, Universität Freiburg, 2005. At ftp://ftp.informatik.uni-freiburg.de/documents/reports/report218/ report00218.ps.gz.

[Koehler, 1998] J. Koehler. Planning under resource constraints. In *Proc. ECAI-98*, pages 489–493.

[Mitchell *et al.*, 1992] D. Mitchell, B. Selman, and H. Levesque. Hard and easy distributions of SAT problems. In *Proc. AAAI-92*, pages 459–465.

[Moskewicz *et al.*, 2001] M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: Engineering an efficient SAT solver. In *Proc. DAC'01*.

[2] The translation implements the multiple-valued variable for each dimension by a set of boolean variables a that cover (only) the relevant case distinctions in that dimension. Binary exclusion clauses of the form $\neg a \vee \neg a'$ ensure that, by unit propagation, only one variable a can be set to 1 at a time.