To Guess or to Think?
Hybrid Algorithms for SAT
(Extended Abstract) *

Irina Rish and Rina Dechter

Information and Computer Science
University of California, Irvine, CA 92717, U.S.A.
{irinar, dechter}@ics.uci.edu http://www.ics.uci.edu/

Complete algorithms for solving propositional satisfiability fall into two main
classes: backtracking search (e.g., the Davis-Putnam Procedure [11]) and resolu-
tion (e.g., the original Davis-Putnam Algorithm [2] and Directional Resolution
[4]). Backtracking may be viewed as a systematic “guessing” of variable assign-
ments, while resolution is inferring, or “thinking”. Experimental results show
that “pure guessing” or “pure thinking” might be inefficient. We propose an ap-
proach that combines both techniques and yields a family of hybrid algorithms
parameterized by a bound on the “effective” amount of resolution allowed. The
idea is to divide the set of propositional variables into two classes: conditioning
variables, which are assigned truth values, and resolution variables, which are re-
solved upon. We report on preliminary experimental results demonstrating that
on certain classes of problems hybrid algorithms are more efficient than either
of their components in isolation.

The well-known Davis-Putnam Procedure (DP) is a backtracking algorithm
enhanced by unit resolution at each level of the search. Directional Resolution
(DR)[4] is a variable-elimination algorithm similar to adaptive-consistency for
constraint satisfaction. Its worst-case time and space complexity is exponential
in induced width, w*, of the interaction graph of a propositional theory. The time
complexity of DP is worst-case exponential in the number of variables, while its
space complexity is linear. However, on average DP is relatively efficient, while
DR’s average complexity is close to its worst-case. Consequently, DR is signifi-
cantly less efficient than DP when applied to uniformly generated 3-cnf’s having
large w*, while outperforming DP by many orders of magnitude when applied
to theories with bounded w* [4]. This time- and space-wise complementary
behavior of the two algorithms prompted the idea of combining DP and DR.

We propose a family of hybrid algorithms, called Dynamic Conditioning +
DR (DCDR), parameterized by a bound, b, that controls the balance between
resolution and backtracking. Given b, the algorithm DCDR(b) selects a subset of
conditioning variables, or cut-set, C_b, such that w* of the resulting (conditional)
theory does not exceed b. The hybrid algorithm searches the space of truth
assignments for the conditioning variables and resolves upon the rest of the

* This work was partially supported by NSF grant IRI-9157636, by the Electrical
Power Research Institute (EPRI) grant RP8014-06, and by Rockwell MICRO grants
UCM-90775 and 95-043.
variables. Dividing the set of variables into the cutset and resolution variables is accomplished during run time, i.e. dynamically. We have also experimented with a static version of the algorithm (for details see the full paper available through http://www.ics.uci.edu/~irinar). We show that the time complexity of both algorithms is $O(exp(c+b))$, where $c$ is the largest cutset size encountered during run time.

We tested DCDR($b$) on uniform $k$-cnfs and on structured problems having bounded $w^*$, such as $(k,m)$-trees. A $(k,m)$-tree is a tree of cliques, each having $k + m$ nodes, where $k$ is the size of intersection between each two neighboring cliques. We observed three different behavior patterns depending on $w^*$ (see Figure 1): 1. on problems having large $w^*$, such as uniform 3-cnfs around the 50% solvable crossover point (the transition region from satisfiable to unsatisfiable problems), the time complexity of DCDR($b$) is similar to DP when $b$ is small (obviously, a bound $b = -1$ does not allow any resolution, making DP equivalent to DCDR(-1)), however, when $b$ increases, the CPU time for DCDR($b$) grows exponentially; 2. theories having very small $w^*$ (such as $(k,m)$-trees with $k \leq 4, m \leq 6$) are easier for DCDR($b$) with a large $b$, since DCDR($b$) coincides with DR for $b \geq w^*$; 3. on $(k,m)$-trees with larger clique size, we observed an intermediate region of $b$'s values yielding a faster algorithm than both DP and DR. The averages for uniform 3-cnfs are computed on 100 problem instances, while for $(k,m)$-trees we ran only 25 experiments per point. We therefore view our results as preliminary. However, they indicate the general promise of the approach.

![Figure 1](image)

We see that $w^*$ provides a reasonable predictor of $b$. When $w^*$ is very large, we choose $b \leq 1$; when $w^*$ is very small (less than 4), we choose large $b$; for intermediate levels of $w^*$ it is better to choose a bounded level of $b$. The algorithms having $b$ in the range of 5 to 8 seem promising, since they behave similarly to DP on uniform instances, to DR for small $w^*$, while for intermediate values of $w^*$ they exploit the benefits of both DP and DR. The hybrid algorithms trade space for time [3], and output a compiled theory from which a portion of the solution set rather than one solution can be generated in linear time.

**References**

This article was processed using the \LaTeX{} macro package with LLNCs style