Resolution Enhanced SLS solver: R+AdaptNovelty$^+$

Duc Nghia Pham[1] **and Anbulagan**[2]
[1]SAFE Program, Queensland Lab, National ICT Australia
[2]L&C Program, Canberra Lab, National ICT Australia
{duc-nghia.pham|anbulagan}@nicta.com.au

1 Introduction

Recent work on Stochastic Local Search (SLS) for the SAT and CSP domains has shown the superior performance of SLS over traditional backtracking algorithms on a broad range of problem instances. In this paper, we report on a technique for enhancing the performance of SLS solvers by incorporating a preprocessing phase in which resolution is used to deduce consequences of the input clauses, exposing hidden structure in the problems which the solvers are then able to exploit.

In the next section we describe the resolution procedure and briefly review some of its uses in SAT solvers. In the subsequent section, we report the resolution enhanced procedure for SLS solvers and outline one enhanced SLS solver, namely R+AdaptNovelty$^+$, based on AdaptNovelty$^+$ [Hoos, 2002].

2 Resolution-based Preprocessing

Resolution is a rule of inference widely used in automated deduction [Quine, 1955; Davis and Putnam, 1960; Robinson, 1965]. In the world of complete search, it is sometimes used to enhance the performance of complete SAT solvers. The first efficient integration of resolution into such a solver was done by Billionnet and Sutter [1992] helping them to solve randomly generated 3-SAT problems with up to 300 variables. Later, Satz [Li and Anbulagan, 1997] used a restricted resolution procedure, adding resolvents of length ≤ 3, as a first phase process before running the complete backtrack search. This implementation gave modest performance gains (around 10%) on random 3-SAT problems, but was very important to the ability of Satz to solve many real-world benchmark problems.

In the incomplete search world, Cha and Iwama [1996] were the first to use a restricted form of resolution procedure called neighbour resolution, which adds new resolvent clauses based on unsatisfied clauses at the local minima and their neighboring clauses. Their study showed that on hard random 3-SAT formulas, the clause weighting strategy is significantly better with neighbour resolution integrated than without. More recently, Fang and Ruml [2004] implemented the same idea along with other techniques in their complete local search solver, where it provides a slight improvement.

In Algorithm 1, we sketch the resolution process implemented in Satz. When two clauses of a CNF formula have

---

Algorithm 1 ComputeResolvents($\mathcal{F}$)

for each clause c_1 of length ≤ 3 in $\mathcal{F}$ **do**
  for each literal l of c_1 **do**
    for each clause c_2 of length ≤ 3 in $\mathcal{F}$ s.t. $\bar{l} \in c_2$ **do**
      Compute resolvent $r = (c_1 \setminus \{l\}) \cup (c_2 \setminus \{\bar{l}\})$;
      if r is empty **then**
        return "unsatisfiable";
      else
        if r is of length ≤ 3 **then**
          $\mathcal{F} := \mathcal{F} \cup \{r\}$;
        end if
      end if
    end for
  end for
end for

---

the property that some variable x_i occurs positively in one and negatively in the other, the resolvent of the clauses is a disjunction of all the literals occuring in the clauses except x_i and $\overline{x_i}$. For example, the clause $(x_2 \vee x_3 \vee \overline{x_4})$ is the resolvent for the clauses $(\overline{x_1} \vee x_2 \vee x_3)$ and $(x_1 \vee x_2 \vee \overline{x_4})$ and is added to the clause set. The new clauses, provided they are of length ≤ 3, can in turn be used to produce other resolvents. The process is repeated until saturation. Duplicate and subsumed clauses are deleted, as are tautologies and any duplicate literals in a clause. It is worth noting that this resolution phase takes polynomial time.

3 Adding Resolution to Modern SLS Solvers

The introduction of GSAT [Selman *et al.*, 1992] and many subsequent SLS variants of WalkSAT [Selman *et al.*, 1994; McAllester *et al.*, 1997] has caused considerable research interest in modelling hard combinatorial problems into SAT and applying SLS solvers to find the solutions. The best contemporary solver in the WalkSAT family is AdaptNovelty$^+$ [Hoos, 2002], which is an automated version of Novelty$^+$ [McAllester *et al.*, 1997]. Like other WalkSAT variants, the performance of Novelty$^+$ critically depends on the setting of its noise parameter, which controls the greediness of the search. AdaptNovelty$^+$ addresses this problem by adaptively tuning its noise level based on the detection of stagnation: it starts with a 0 noise level. If no improvement in the objective evaluation value, which is the count of unsatisfied clauses, has been made after a number of flips, the noise level is increased. As soon as the value of the objective function is improved over its value at the last change of the noise level, the noise level is reduced. Experimental results have shown that this adaptive noise mechanism also works well with other WalkSAT variants [Hoos, 2002].

Our main observation is that SLS algorithms benefit markedly from resolution preprocessing. We therefore propose a two-phase SLS algorithm that further improves the performance of SLS solvers by having extra information from resolution. In the first stage, such a two-phase algorithm calls the ComputeResolvents procedure in Algorithm 1 to add resolvent clauses to $\mathcal{F}$ and also remove duplicate clauses, the tautologies, and duplicate literals in a clause. It then run the SLS solver on the resulting clause set $\mathcal{F}_r$.

4 Contest Implementation

For the SAT 2007 competition, R+AdaptNovelty+ was implemented based on the efficient data structure used in the UBCSAT framework [Tompkins and Hoos, 2004]. The resolution process was derived from the source code of Satz [Li and Anbulagan, 1997].

For R+AdaptNovelty$^+$, parameters were set to the default values of the UBCSAT version of AdaptNovelty$^+$ [Tompkins and Hoos, 2004], where $(\Phi, \Theta) = (|\frac{1}{5}, \frac{1}{6}|)$.

Acknowledgments

The authors would like to acknowledge the financial support of National ICT Australia (NICTA) and the Queensland government. NICTA is funded through the Australian Government's *Backing Australia's Ability* initiative and also through the Australian Research Council.

References

[Billionnet and Sutter, 1992] Alain Billionnet and Alain Sutter. An efficient algorithm for 3-satisfiability problem. *Operations Research Letters*, 12:29–36, 1992.

[Cha and Iwama, 1996] Byungki Cha and Kazuo Iwama. Adding new clauses for faster local search. In *Proceedings of the 13th AAAI*, pages 332–337, 1996.

[Davis and Putnam, 1960] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.

[Fang and Ruml, 2004] Hai Fang and Wheeler Ruml. Complete local search for propositional satisfiability. In *Proceedings of the 19th AAAI*, pages 161–166, 2004.

[Hoos, 2002] Holger H. Hoos. An adaptive noise mechanism for WalkSAT. In *Proceedings of the 18th AAAI*, pages 655–660, 2002.

[Li and Anbulagan, 1997] Chu Min Li and Anbulagan. Look-ahead versus look-back for satisfiability problems. In *Proceedings of the 3rd CP*, pages 341–355, 1997.

[McAllester *et al.*, 1997] D. A. McAllester, Bart Selman, and Henry A. Kautz. Evidence for invariants in local search. In *Proceedings of the 14th AAAI*, pages 321–326, 1997.

[Quine, 1955] W. V. Quine. A way to simplify truth functions. *American Mathematical Monthly*, 62:627–631, 1955.

[Robinson, 1965] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.

[Selman *et al.*, 1992] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the 10th AAAI*, pages 440–446, 1992.

[Selman *et al.*, 1994] Bart Selman, Henry A. Kautz, and B. Cohen. Noise strategies for improving local search. In *Proceedings of the 12th AAAI*, pages 337–343, 1994.

[Tompkins and Hoos, 2004] Dave A.D. Tompkins and Holger H. Hoos. Warped landscapes and random acts of SAT solving. In *Proceedings of the 8th International Symposium on Artificial Intelligence and Mathematics*, 2004.