gNovelty$^+$

Duc Nghia Pham and Charles Gretton
SAFE Program, Queensland Research Lab, NICTA
{duc-nghia.pham, charles.gretton}@nicta.com.au

1 Preface

We describe a new stochastic local search (SLS) procedure gNovelty$^+$ for finding satisfying models of satisfiable propositional CNF formulae. This draws on the strengths of the two SLS solvers which placed first and second in the random category of the 2005 SAT competition. In particular, gNovelty$^+$ draws on features of R+AdaptNovelty$^+$ [Anbulagan *et al.*, 2005] that make it effective in random $>$ 3-SAT problems, and features of G^{2}WSAT [Li and Huang, 2005] that make it effective for 3-SAT. However, gNovelty$^+$ makes no use of a resolution-based preprocessing step as R+AdaptNovelty$^+$ does. In addition, gNovelty$^+$ uses clause weighting to gain more efficiency.

This abstract is organised as follows: we review the background of R+AdaptNovelty$^+$ and G^{2}WSAT. We then review the performance of those procedures, analyse circumstances where one outperforms the other, and identify the core techniques behind their success. Finally, we describe gNovelty$^+$ and the technical settings of its contest implementation.

2 Background

2.1 G^{2}WSAT

For a little while Novelty [McAllester *et al.*, 1997] was one of the best solvers in the WalkSAT family [Selman *et al.*, 1994] of SLS procedures and was able to solve many hard problems better than systematic solvers. One key problem with Novelty is its deterministic variable selection: it may loop indefinitely and fail to return a solution even where one exists [Hoos, 1999; Li and Huang, 2005].[1] The first practical solution to this problem was proposed by Hoos [1999]. The resulting Novelty$^+$ solver performs a random walks with a probability wp on top of the Novelty procedure. More recently Li and Huang [2005] revisited this problem and proposed a more diversified heuristic to weaken the determinism in Novelty. The new Novelty^{++} solver selects the least recently flipped variable for the next move with a *diversification probability dp*, otherwise it performs as Novelty. Li and Huang [2005] further improved Novelty^{++} by integrating it with a new gradient-based greedy heuristic. The resulting G^{2}WSAT solver (depicted in the left hand side of Figure 3) always selects the best promising variable by score for the next move. For example, if the objective function is the count of current false clauses, then a promising variable is one that, if flipped, will reduce the number of false clauses. If there is more than one variable with the best score, G^{2}WSAT selects the least recently flipped one. If the search hits a local minimum, G^{2}WSAT performs as Novelty^{++} until it escapes that local minimum.

2.2 R+AdaptNovelty$^+$

The performance of every WalkSAT variant critically depends on the setting of its noise parameter which controls the level of greediness (resp. randomness) of the search [McAllester *et al.*, 1997; Hoos, 2002]. Hoos [2002] proposed a (kind-of) parameterless version of WalkSAT that adaptively tunes the noise level of a WalkSAT solver based on the automatic detection of search stagnation. For example, AdaptNovelty$^+$, the automated version of Novelty$^+$, starts with a 0 noise level – i.e. the solver is completely greedy in searching for the next move. If no improvement in the objective is made after a number of flips, the noise level is increased to allow more non-greedy moves to be performed. 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 to make the search more greedy. Hoos [2002] demonstrated experimentally that his adaptive noise mechanism is effective with Novelty$^+$ and other WalkSAT variants.

In 2005, Anbulagan *et al.* [2005] introduced a two-phase SLS solver called R+AdaptNovelty$^+$, which further improves the performance of AdaptNovelty$^+$ by using a resolution preprocessor to derive extra information from the input. In particular, R+AdaptNovelty$^+$ applies a restricted resolution procedure to all clauses of length ≤ 3 from the input. This process adds resolvent clauses of length ≤ 3 to the problem, and also removes duplicate clauses, the tautologies, and literals that appear twice in a single clause. It then runs AdaptNovelty$^+$ on the resulting problem.

[1] Novelty deterministically selects the next move from the two best variables of a randomly selected false clause [McAllester *et al.*, 1997]. Hoos [1999] gave an example instance that is satisfiable but Novelty loops indefinitely and is unable to find a solution for that instance regardless of the noise parameter setting.

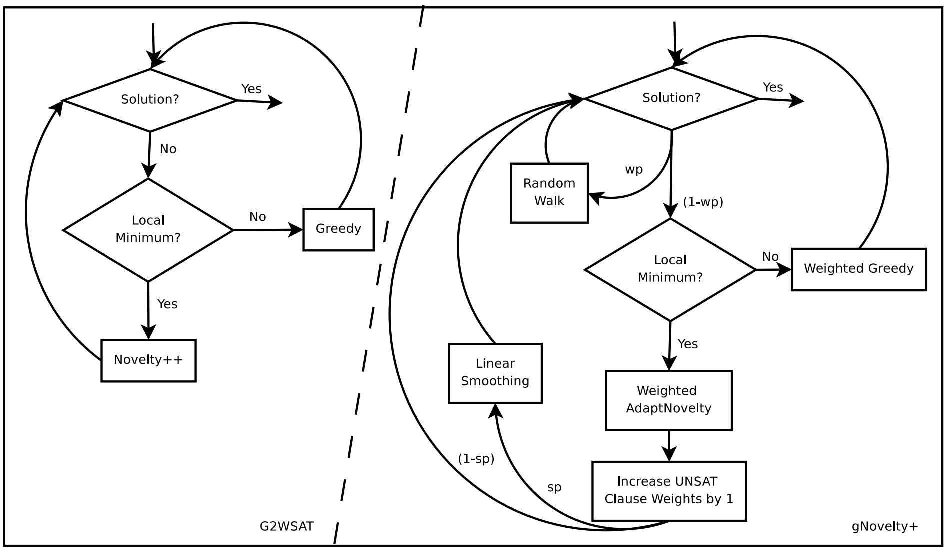

Figure 1: Flow-chart comparison between the two procedures G^2WSAT and gNovelty$^+$.

3 gNovelty$^+$

3.1 Lessons learnt

Examining the results from the 2005 SAT competition one observes the following: although G^2WSAT came second in the random SAT category, it was the best solver for random 3-SAT instances. Neither R+AdaptNovelty$^+$ or AdaptNovelty$^+$ is competitive with G^2WSAT on those instances, thus we suppose the performance of the latter is due to its greedy behaviour. This supposition is consistent with the results reported by Li and Huang [2005] when they compared it with Novelty$^+$.

On the other hand, R+AdaptNovelty$^+$ outperformed G^2WSAT on the 5-SAT and 7-SAT competition instances. This was not due to the resolution preprocessor employed by R+AdaptNovelty$^+$ because, in such instances, it simply reorders the occurrence of literals in a clause. We also empirically find that G^2WSAT relies heavily on its Novelty^{++} component on hard random 5-SAT and 7-SAT instances. Finally we observe that AdaptNovelty$^+$ is generally a more effective Novelty variant for use by G^2WSAT than Novelty^{++}.

3.2 The algorithm

In composing gNovelty$^+$ we replaced the Novelty^{++} heuristic in G^2WSAT with the AdaptNovelty heuristic to enhance its performance on the 5-SAT and 7-SAT instances. As clause weighting techniques [Hutter *et al.*, 2002; Thornton *et al.*, 2004] are very effective in greedily guiding the search toward the solution, we further incorporate this technique into our new solver. The resulting gNovelty$^+$ solver is sketched out in Algorithm 1 and depicted diagrammatically in Figure 3.

At every search step, gNovelty$^+$ selects the most promising variable that is also the least recently flipped, based on our *weighted* objective function. Our objective is to minimise the sum of weights of all false clauses. If no such promising variable exists, the next variable is selected using a heuristic based on AdaptNovelty that utilises the *weighted* objective function. After the Novelty step, gNovelty$^+$ increase the weights of all current false clauses by 1.[2] In order to keep the control of the level of greediness of the search flexible, we also incorporates into gNovelty$^+$ a new linear version of the probabilistic weight smoothing from SAPS [Hutter *et al.*, 2002]. Every time gNovelty$^+$ updates its clause weights, with a *smoothing probability* sp the weights of all *weighted clauses* (a clause is weighted if its weight is greater than one) are subject to a reduction of 1. Finally, we also added a probabilistic walk heuristic (i.e. the *plus* heuristic from Hoos [1999]) to gNovelty$^+$ to further improve the balance between the level of randomness (resp. greediness) of the search.

Algorithm 1 gNovelty$^+$(F)

for $try = 1$ to $maxTries$ **do**
  initialise the weight of each clause to 1;
  randomly generate an assignment A;
  for $step = 1$ to $maxSteps$ **do**
    if A satisfies F **then**
      return A as the solution;
    else
      if within a walking probability wp **then**
        randomly select a variable x that appears in a false clause;
      else if there exist *promising* variables **then**
        greedily select a promising variable x, breaking tie by selecting the least recently flipped one;
      else
        select a variable x according to the *weighted AdaptNovelty* heuristic;
        update the weights of false clauses;
        with probability sp smooth the weights of all clauses;
      end if
      update A with the flipped value of x;
    end if
  end for
end for
return 'no solution found';

[2]We decided to use the additive weight increase at each local minimum as it is cheaper to maintain than its counterpart multiplicative weighting [Thornton *et al.*, 2004].

4 Contest Implementation

For the 2007 SAT competition, the parameter sp of gNovelty$^+$ is fixed at $.4$ for the 3-SAT problems and at 1 for other problems. Also, wp was always set to 0.01.

Acknowledgements

We thankfully acknowledge the financial support from National ICT Australia (NICTA) and the Queensland government. National ICT Australia is funded through the Australian Government's *Backing Australia's Ability* initiative and partly through the Australian Research Council.

References

[Anbulagan *et al.*, 2005] Anbulagan, Duc Nghia Pham, John Slaney, and Abdul Sattar. Old resolution meets modern SLS. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, pages 354–359, 2005.

[Hoos, 1999] Holger H. Hoos. On the run-time behaviour of stochastic local search algorithms for SAT. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 661–666, 1999.

[Hoos, 2002] Holger H. Hoos. An adaptive noise mechanism for WalkSAT. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02)*, pages 635–660, 2002.

[Hutter *et al.*, 2002] Frank Hutter, Dave A.D. Tompkins, and Holger H. Hoos. Scaling and probabilistic smoothing: Efficient dynamic local search for SAT. In *Proceedings of the Eighth International Conference on Principles and Practice of Constraint Programming (CP-02)*, pages 233–248, 2002.

[Li and Huang, 2005] Chu Min Li and Wen Qi Huang. Diversification and determinism in local search for satisfiability. In *Proceedings of the Eighth International Conference on Theory and Applications of Satisfiability Testing (SAT-05)*, pages 158–172, 2005.

[McAllester *et al.*, 1997] David A. McAllester, Bart Selman, and Henry A. Kautz. Evidence for invariants in local search. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 321–326, 1997.

[Selman *et al.*, 1994] Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 337–343, 1994.

[Thornton *et al.*, 2004] John R. Thornton, Duc Nghia Pham, Stuart Bain, and Valnir Ferreira Jr. Additive versus multiplicative clause weighting for SAT. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-04)*, pages 191–196, 2004.