Switching Between Two Adaptive Noise Mechanisms in Local Search for SAT

Wanxia Wei[1] and Chu Min Li[2]

[1] Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, E3B 5A3
wanxia.wei@unb.ca *
[2] MIS, Université de Picardie Jules Verne 33 Rue St. Leu, 80039 Amiens Cedex 01, France
chu-min.li@u-picardie.fr

1 New Mechanism for Adaptively Adjusting Noise

The adaptive noise mechanism was introduced in [1] to automatically adjust noise during the search. We refer to this mechanism as Hoos's noise mechanism. This mechanism adjusts noise based on search progress and applies the adjusted noise to variables in any clause in a search step.

We propose a new mechanism for adaptively adjusting noise during the search. This mechanism uses the history of the most recent consecutive falsifications of a clause. During the search, for the variables in each clause, we record both the variable that most recently falsifies this clause and the number of the most recent consecutive falsifications of this clause due to the flipping of this variable. For a clause *c*, we use $var\_fals[c]$ to denote the variable that most recently falsifies *c* and use $num\_fals[c]$ to denote the number of the most recent consecutive falsifications of *c* due to the flipping of this variable. Assume that *c* is falsified most recently by variable *x* in *c* and so far *x* has consecutively falsified clause *c* *m* times. So, for *c*, $var\_fals[c] = x$ and $num\_fals[c] = m$. If *c* is falsified again, there are two cases. One is that *x* falsifies *c* again. In this case, $var\_fals[c]$ is still *x*, and $num\_fals[c]$ becomes *(m + 1)*. The other is that another variable *y* in *c* falsifies *c*. In this case, $var\_fals[c]$ becomes *y*, and $num\_fals[c]$ becomes *1*. Assume that clause *c* is the currently selected unsatisfied clause and that a variable in *c* will be chosen to flip. We use $best\_var$ to represent the best variable in clause *c* measured by the scores of all variables in *c*. If $best\_var$ is not $var\_fals[c]$, this mechanism sets noise to its lowest value 0.00 in order to choose $best\_var$ to flip. If $best\_var$ is $var\_fals[c]$, this mechanism determines noise according to $num\_fals[c]$. Specifically, the higher $num\_fals[c]$ is, the higher the noise value is.

Our mechanism for adjusting noise is different from Hoos's noise mechanism in two respects. First, our mechanism uses the history of the most recent consecutive falsifications of a clause due to the flipping of one variable in this clause, while Hoos's noise mechanism observes the improvement in the objective function value. Second, the noise adjusted by our mechanism is clause-specific, whereas the noise adjusted by Hoos's noise mechanism is not.

2 New Local Search Algorithm TNM

Variable weighting was introduced in [4]. The weight of a variable x, $vw[x]$, is initialized to 0 and is updated and smoothed each time x is flipped, using the following formula:

$$vw[x] = (1 - s)(vw[x] + 1) + s \times t \tag{1}$$

where s is a parameter and $0 \leq s \leq 1$, and t denotes the time when x is flipped, i.e., t is the number of search steps since the start of the search [4].

If all variables in all clauses have roughly equal chances of being flipped, all variables should have approximately equal weights. In this case, the same noise can be applied to any variable in any clause at a search step. Otherwise, our proposed mechanism can be used to adjust noise for the variables in each specific clause in order to break stagnation.

* The first author can be reached via e-mail at weiwanxia@gmail.com after graduation.

A switching criterion, namely, the evenness or unevenness of the distribution of variable weights, was introduced in [6]. We propose a new local search algorithm called TNM (Two Noise Mechanism), which switches between Hoos's noise mechanism and our proposed noise mechanism according to this criterion. This algorithm is described in Fig. 1. Hoos's noise mechanism was integrated in G^2WSAT [2], resulting in $adaptG^2WSAT$ [3]. The local search algorithm $adaptG^2WSAT+$ [5] was improved from $adaptG^2WSAT$. We integrate our proposed noise mechanism to G^2WSAT [2] and obtain $adaptG^2WSAT'$. In Fig. 1, parameter γ ($\gamma > 1$) determines whether the distribution of variable weights is uneven. TNM sets γ to its default value 10.0. Parameters $p1$ and $p2$ represent the noise values adjusted by Hoos's noise mechanism and by our proposed mechanism, respectively. TNM updates variable weights using Formula 1.

Algorithm: TNM(SAT-formula $\mathcal{F}$)

$A \leftarrow$ randomly generated truth assignment;
for each clause j **do** initialize $var\_fals[j]$ and $num\_fals[j]$ to -1 and 0, respectively;
for each variable x **do** initialize $flip\_time[x]$ and $var\_weight[x]$ to 0;
initialize $p1$, wp, s, max_weight, and ave_weight to 0; initialize dp to 0.05;
store promising decreasing variables in stack *DecVar*;
for *flip*$\leftarrow 1$ **to** *Maxsteps* **do**
    if A satisfies $\mathcal{F}$ **then return** A;
    if $max\_weight \geq \gamma \times ave\_weight$
    then
        if there is no promising decreasing variable
        then
            randomly select an unsatisfied clause c;
            adjust $p2$ for variables in c according to $var\_fals[c]$ and $num\_fals[c]$;
        $y\leftarrow$*heuristic adapt*G^2WSAT'($p2$, *dp*);
    else $y\leftarrow$*heuristic adapt*G^2WSAT+($p1$, *wp*);
    $A\leftarrow A$ with y flipped;
    if flippin of y falsifies a clause j **then** update $var\_fals[j]$ and $num\_fals[j]$;
    adjust $p1$ according to Hoos's noise mechanism; $wp = p1/10$;
    update $flip\_time[y]$, $var\_weight[y]$, max_weight, ave_weight, and *DecVar*;
return Solution not found;

Fig. 1. Algorithm TNM

Our first implementation of the proposed noise mechanism in algorithm TNM is simple. Assume that the currently selected unsatisfied clause is falsified most recently by variable *x* and so far *x* has consecutively falsified this clause *m* times. If the best variable measured by the scores of all variables in this clause is not x, we set noise $p2$ to 0.00. Otherwise, we set $p2$ to a reasonable value according to m.

References

1. H. H. Hoos. An Adaptive Noise Mechanism for WalkSAT. In *Proceedings of AAAI-2002*, pages 655–660. AAAI Press, 2002.
2. C. M. Li and W. Q. Huang. Diversification and Determinism in Local Search for Satisfiability. In *Proceedings of SAT-2005*, pages 158–172. Springer, LNCS 3569, 2005.
3. C. M. Li, W. Wei, and H. Zhang. Combining Adaptive Noise and Look-Ahead in Local Search for SAT. In *Proceedings of SAT-2007*, pages 121–133. Springer, LNCS 4501, 2007.
4. S. Prestwich. Random Walk with Continuously Smoothed Variable Weights. In *Proceedings of SAT-2005*, pages 203–215. Springer, LNCS 3569, 2005.
5. W. Wei, C. M. Li, and H. Zhang. Deterministic and Random Selection of Variables in Local Search for SAT. http://www.satcompetition.org/2007/contestants.html.

6. W. Wei, C. M. Li, and H. Zhang. Criterion for Intensification and Diversification in Local Search for SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 4:219–237, June 2008.