Definitions

Chuan Luo[1,2], Holger H. Hoos[2], and Shaowei Cai[3]

[1] Microsoft Research, China
[2] Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
[3] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China
chuan.luo@microsoft.com, hh@liacs.nl, caisw@ios.ac.cn

1 Definitions of Variable Properties

1.1 score

For a variable x, the *score* of x, denoted $\mathit{score}(x)$, is the increment in the number (or total weight if using clause weighting scheme) of satisfied clauses by flipping x [7].

A variable is a *decreasing* variable if its score is positive.

1.2 age

For a variable x, the *age* of x, denoted $\mathit{age}(x)$, is the number of steps that has occured since x's last flip [5].

1.3 subscore

A clause is δ-satisfied if there are exactly δ true literals in that clause under the current assignment [4].

For a variable x, the *subscore* of x, denoted $\mathit{subscore}(x)$, is defined as $\mathit{submake}(x) - \mathit{subbreak}(x)$, where $\mathit{submake}(x)$ is the number of 1-satisfied clauses that would become 2-satisfied by flipping x, and $\mathit{subbreak}(x)$ is the number of 2-satisfied clauses that would become 1-satisfied by flipping x [3].

1.4 hscore

For a variable x, the *hscore* of x is defined as $\mathit{hscore}(x) = \mathit{score}(x) + \lfloor \mathit{subscore}(x)/\mathit{hscore\_d} \rfloor + \lfloor \mathit{age}(x)/\mathit{hscore\_\beta} \rfloor$, where $\mathit{hscore\_d}$ and $\mathit{hscore\_\beta}$ are positive integers [3].

1.5 *hscore₂*

For a variable x, the hscore_2 of x is defined as $\mathit{hscore}_2(x) = \mathit{subscore}(x) + \lfloor \mathit{age}(x)/\mathit{hscore}_2\_\gamma \rfloor$, where hscore_2_γ is a positive integer [1].

2 Definitions in Configuration Checking

In the context of SLS for solving SAT, different definitions of context give rise to different CC strategies – notably, Clause-states-based configuration checking (CSCC) [6] and neighbouring-variables-based configuration checking (NVCC) [4]:

- In CSCC, the context of a variable x is defined as a vector consisting of clause states of all clauses containing x. A variable x is *clause-states-based configuration-changed decreasing* (CSD) if (1) $score(x) > 0$ and (2) at least one clause containing x has changed its clause state (from satisfied to unsatisfied or vice versa) since x was last flipped. During the search process, the set of all CSD variables is denoted as $CSDvars$.
- In NVCC, the context of variable x is given by a vector consisting of the truth values assigned to all its neighbouring variables (*i.e.,* those variables which differ from x and appear in at least one clause where x appears). A variable x is *neighbouring-variables-based configuration-changed decreasing* (NVD) if (1) $score(x) > 0$ and (2) one of the neighbouring variables of x has changed its truth value (from *True* to *False* or vice versa) since x was last flipped. During the search process, the set of all NVD variables is denoted as $NVDvars$.

3 Definitions in the Aspiration Mechanism

The key concept underlying this mechanism is the *significant decreasing* (SD) property of a variable [2]:

A variable is SD if $score(x) > \overline{w}$, where $\overline{w}$ is the average weight of the clauses in the given CNF formula (if no clause weighting scheme is activated, a variable is SD if $score(x) > 1$). During the search process, the set of all SD variables is denoted as $SDvars$.

References

1. Cai, S., Luo, C., Su, K.: Scoring functions based on second level score for k-SAT with long clauses. Journal of Artificial Intelligence Research **51**, 413–441 (2014)
2. Cai, S., Su, K.: Configuration checking with aspiration in local search for SAT. In: Proceedings of AAAI 2012. pp. 434–440 (2012)
3. Cai, S., Su, K.: Comprehensive score: Towards efficient local search for SAT with long clauses. In: Proceedings of IJCAI 2013. pp. 489–495 (2013)
4. Cai, S., Su, K.: Local search for Boolean satisfiability with configuration checking and subscore. Artificial Intelligence **204**, 75–98 (2013)
5. Gent, I.P., Walsh, T.: Towards an understanding of hill-climbing procedures for SAT. In: Proceedings of AAAI 1993. pp. 28–33 (1993)
6. Luo, C., Cai, S., Su, K., Wu, W.: Clause states based configuration checking in local search for satisfiability. IEEE Transactions on Cybernetics **45**(5), 1014–1027 (2015)
7. Selman, B., Levesque, H.J., Mitchell, D.G.: A new method for solving hard satisfiability problems. In: Proceedings of AAAI 1992. pp. 440–446 (1992)