Adv. Studies Theor. Phys., Vol. 7, 2013, no. 8, 363 - 366
HIKARI Ltd, www.m-hikari.com

GSAT with Adaptive Score Function

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

GSAT is a well-known satisfiability search algorithm. In this paper we consider a modification of GSAT. In particular, we consider an adaptive score function.

PACS: 02.70.Rr

Keywords: adaptive score function, GSAT, satisfiability

Investigation of efficient satisfiability algorithms and encoding different hard problems as instances of satisfiability has caused considerable interest recently (see e.g. [1, 2]). GSAT is a well-known satisfiability algorithm [3]. Note that GSAT applies only to clausal formulas. The notion of score plays a key role in GSAT. Although, some modification of the score function are considered, usually, the score function is the number of clauses of formula which are falsified by the truth assignment (see e.g. [4]). In this paper we consider a modification of the score function for clausal formulas with different types of clauses.

At first, we consider a general schema for standard GSAT (see Figure 1). We use the notation from [5]. Let $f(z[1], \ldots, z[n])$ be a CNF. Let T be a truth assignment for the variables of f. The function $score(T, f)$ is the number of clauses of f which are falsified by T. GSAT performs an search for a satisfying truth assignment for the variables of f, starting from a random assignment provided by $initial(f)$. The successive assignment is obtained by inverting the truth value of one single variable V in T. The value of V is chosen to minimize the score. Let T_i be the sequence obtained from T by inverting the value of ith variable V_i. The function $hill-climb(f,T)$ returns the set $Poss-flips$ of the variables V_r which minimize $score(T_r,f)$. The function $pick(Poss-flips)$ chooses randomly one of elements of $Poss-flips$. The function $flip(V,T)$ returns T with V's value inverted. The function $UpdateScores(f,V)$ updates score.

```
procedure GSAT(f)
    for j := 1 to Max − tries do
        T := initial(f)
        for k := 1 to Max − flips do
            if T |= f then return T
                else Poss − flips := hill − climb(f, T)
                    V := pick(Poss − flips)
                    T := flip(V, T)
                    UpdateScores(f, V)
        end
    end
    return "no satisfying assignment found".
```

Figure 1: A general schema for GSAT.

The function $score(T,f)$ is the number of clauses of f which are falsified by T. In general, there are a number of different types of clauses. For instance, there is some difference between clauses x and $y \vee z$. But, without additional infirmation, we can not give reasonable score of such difference. In particular, if $x = y = z = u = v = 0$,

$$x \wedge (y \vee z) \wedge (\neg x \vee u) \wedge (\neg x \vee v) \wedge (y \vee u), \quad (1)$$

$$x \wedge (y \vee z) \wedge (\neg x \vee u) \wedge (\neg y \vee u) \wedge (\neg y \vee v) \wedge (\neg z \vee u) \wedge (\neg z \vee v), \quad (2)$$

then we need modify two values of variables (u and v) for x and only one value of variable (u) for $y \vee z$ in formula (1), but we need modify two values of variables (u and v) for $y \vee z$ and only one value of variable (u) for x in formula (2). So, we can assume that the score of x higher than the score of $y \vee z$, for formula (1), and the score of x lower than the score of $y \vee z$, for formula (2).

Let $\#occ(k,f,z[i])$ be the number of positive occurrences of $z[i]$ in clauses of type $x_1 \vee \ldots \vee x_k$, $\#occ(k,f,\neg z[i])$ be the number of occurrences of $\neg z[i]$ in clauses of type $x_1 \vee \ldots \vee x_k$. Let $S(T,f)$ be the set of clauses of f which are falsified by T. We assume that

$$score(T,f) = \sum_{x_1 \vee \ldots \vee x_p \in S(T,f)} \sum_{1 \le i \le q, 1 \le j \le p} (\alpha(\#occ(i,f,x_j)) + \beta(\#occ(i,f,\neg x_j))).$$

We use a genetic algorithm for prediction of values of α and β.

Now, we consider a special class of formulas 3-2-CNF. We assume that any clause of formula from this class belongs the set

$$\{z[i] \vee z[j] \vee z[l], \neg z[i] \vee \neg z[j] \mid i, j, l \in N\}. \quad (3)$$

Theorem. *For any 3-CNF f, there is a formula g such that any clause of g belongs the set (3) and f is satisfiable if and only if g is satisfiable.*

Proof. It is easy to see that $x_1 \vee x_2 \vee \neg x_3$ is satisfiable if and only if $(x_1 \vee x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4)$ is satisfiable. □

In view of theorem, we can be sure that the class 3-2-CNF is sufficiently general. In our experiments, we consider GSAT with standard score function and GSAT with our score function (GSAT-ASF) for CNFs, 3-CNFs, and 3-2-CNFs. Selected experimental results are given in Table 1.

	3-CNF	3-2-CNF	CNF
GSAT	86 %	81 %	79 %
GSAT-ASF, $G = 10^4$	91 %	89 %	81 %
GSAT-ASF, $G = 10^5$	92 %	94 %	85 %
GSAT-ASF, $G = 10^6$	93 %	95 %	88 %

Table 1: A number of solved formulas for GSAT and GSAT-ASF where G is a number of generations of genetic algorithm.

ACKNOWLEDGEMENTS. The work was partially supported by Analytical Departmental Program “Developing the scientific potential of high school” 8.1616.2011.

References

[1] A. Gorbenko and V. Popov, SAT Solvers for the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012), 1235-1238.

[2] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, 9 (2012), 429-441.

[3] B. Selman, H. Levesque, and D. Mitchell, A new method for solving hard satisfiability problems, *Proceedings of the Tenth National Conference on Artificial Intelligence*, (1992), 440-446.

[4] R. Sebastiani, Applying GSAT to non-clausal formulas, *Journal of Artificial Intelligence Research*, 1 (1993), 309-314.

[5] I.P. Gent and T. Walsh, Towards an Understanding of Hill-Climbing Procedures for SAT, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, (1993), 28-33.

Received: February 12, 2013