GLUEMINISAT2.2.5

Hidetomo NABESHIMA[†] Koji IWANUMA[†] Katsumi INOUE[‡]

[†]University of Yamanashi, Japan
[‡]National Institute of Informatics, Japan
{nabesima,iwanuma}@yamanashi.ac.jp
ki@nii.ac.jp

1 Introduction

GLUEMINISAT is a SAT solver based on *literal blocks distance* (LBD) proposed by Audemard and Simon [2] which is an evaluation criteria to predict learnt clauses quality in CDCL solvers. The effectiveness of LBD was shown in their SAT solver GLUCOSE at the latest SAT competition. GLUEMINISAT uses a slightly restricted concept of LBD, called *strict LBD*, and a dynamic restart strategy based on local averages of decision levels and LBDs of learnt clauses.

CDCL solvers learn clauses from conflicts during search. In order to prove unsatisfiability of a SAT instance, it is important to acquire learnt clauses which will lead to a refutation. For this purpose, the following two techniques are required: (1) how to evaluate "good" learnt clauses, and (2) how to get such good learnt clauses. GLUEMINISAT uses the strict LBD measure as an evaluation criteria for (1), and uses the dynamic restart strategy for (2), which intends to decrease decision levels and to get learnt clauses with small LBDs.

The experimental results show GLUEMINISAT is strong in proof of unsatisfiability of SAT instances rather than satisfiability. GLUEMINISAT has some successful results in proving the optimality of known bounds for a open problem of finding optimal covering arrays [3] and in improving known lower/upper bounds for some hard job shop scheduling problems [4].

The remainder of this paper is organized as follows: Section 2 introduces the (strict) LBD measure. Section 3 describes a restart strategy of GLUEMINISAT. Section 4 shows the experimental results. Section 5 concludes with this work.

2 Literal Block Distance

The literal blocks distance (LBD) is proposed by Audemard and Simon [2] in order to evaluate learnt clauses quality in CDCL solvers. A *block* is defined as all literals which are assigned at the same decision level. Such literals have some relationship with each other since they are assigned on the same condition, and are expected that they are appeared the repeated appearance during search. A learnt clause is evaluated by the number of blocks in the clause.

Definition 1 (Literals Blocks Distance (LBD)) [2] *Given a clause C, and a partition of its literals into n subsets according to the current assignment, s.t. literals are partitioned w.r.t their decision level. The LBD of C is exactly n.*

Especially, a clause C whose LBD is two is called a *glue clause* which has a role to connect two blocks. The LBD of a clause is computed when the leant clause is produced. If a clause C is learned by the first UIP schema, then C contain one literal of the last decision level (it is the first UIP). GLUCOSE preserves every glue clauses which are never deleted during search. The LBDs of clauses are recomputed when they are used for unit propagations, and updated if the LBDs become smaller. This update process is important to get many glue clauses.

GLUEMINISAT uses a slightly restricted concept of LBD, called *strict LBD*. The purpose is to avoid generating glue clauses whose every block consists of two or more literals. We consider that a glue clause that has no unit block is less useful than other ones, since a glue clause that consists of non-unit blocks does not invoke a unit propagation even if one block are appeared.

Definition 2 (Strict LBD) *Let C be a clause and n LBD of C. If C has an unit literal block which consists of only one literal, then strict LBD of C is n. Otherwise, strict LBD of C is not defined.*

The number of glue clauses produced by this measure is less than the original LBD measure. Therefore, GLUEMINISAT preserves clauses whose strict LBDs are less than or equal to three.

Table 1: The number of solved instances

	GLUCOSE	MINISAT	GLUEMINISAT
# of solved	133	141	161
SAT / UNSAT	52 / 81	60 / 81	61 / 100

3 Restart Strategy

Interestingly, the LBD measure is incompatible with the well known Luby restart strategy [5]. Fig 1 show the experimental results of MINISAT2.2 and MINISAT2.2 with LBD for the application category of SAT 2009 competition. MINISAT2.2 uses the Luby restart strategy. MINISAT2.2 with LBD is worse than MINISAT2.2.

In order to utilize the LBD measure, it is very important to acquire good learnt clauses. GLUEMINISAT uses a dynamic restart strategy: if one of the following conditions is satisfied, then a restart is forced.

1. an average of *decision levels* over the last 50 conflicts is greater than the global average, or
2. an average of *LBDs* over the last 50 conflicts is greater than the global average × 0.8.

The former was proposed in the system description of GLUCOSE1.0 [1], but the latter one was used in the source code of GLUCOSE1.0. GLUEMINISATuses both of them. This restart strategy intends to decrease decision levels and to get learnt clauses with small LBD values.

4 Experimental Results

GLUEMINISAT is developed based on MINISAT2.2, and implemented the strict LBD measure and the dynamic restart strategy. GLUEMINISAT can be executed as MINISAT2.2 or GLUCOSE by specifying the command line option `-minisat` or `-glucose`, respectively.

Fig 1 and Table 1 show the experimental results for the application category of SAT 2009 competition. Every solver does not use clause simplification. The experiments were conducted on a Core Duo (1.66GHz) machine on Mac OS X 10.5 with 2GB memory. GLUEMINISAT shows good performance compared with GLUCOSE1.0 and MINISAT2.2, and is strong for unsatisfiable instances.

5 Conclusion

The LBD measure is useful to evaluate learnt clause quality, but it requires an acquisition mechanism for good learnt clauses. GLUEMINISAT uses the strict LBD to evaluate better clauses and the dynamic restart strategy to decrease decision levels and to get better clauses.

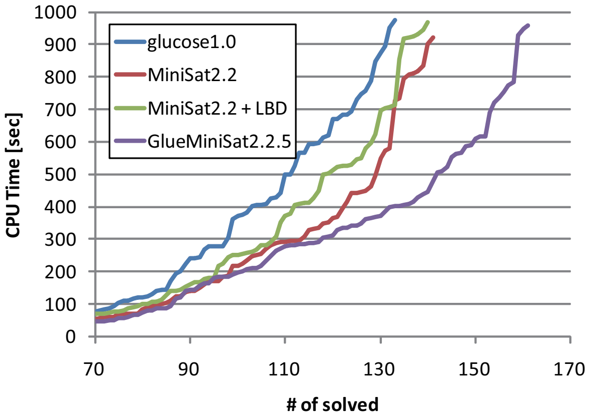

Figure 1: The experimental results for application category of SAT 2009 competition

Acknowledgments

We thank authors of MINISAT which is really an easy extensible SAT solver, and authors of GLUCOSE which gave the very useful criteria of learnt clauses. This research is supported in part by Grant-in-Aid for Scientific Research (No. 20240003) from Japan Society for the Promotion of Science.

References

[1] Gilles Audemard and Laurent Simon. GLUCOSE: a solver that predicts learnt clauses quality. `http://www.cril.univ-artois.fr/SAT09/solvers/booklet.pdf`, 2009. SAT Competition 2009 Solver Description.

[2] Gilles Audemard and Laurent Simon. Predicting learnt clauses quality in modern SAT solvers. In *Proceedings of IJCAI-2009*, pages 399–404, 2009.

[3] Mutsunori Banbara, Haruki Matsunaka, Naoyuki Tamura, and Katsumi Inoue. Generating combinatorial test cases by efficient sat encodings suitable for cdcl sat solvers. In *Proceedings of LPAR-17*, pages 112–126, 2010.

[4] Miyuki Koshimura, Hidetomo Nabeshima, Hiroshi Fujita, and Ryuzo Hasegawa. Solving open job-shop scheduling problems by sat encoding. *IEICE Transactions on Information and Systems*, 93(8):2316–2318, 2010.

[5] Michael Luby, Alistair Sinclair, and David Zuckerman. Optimal speedup of las vegas algorithms. *Information Processing Letters*, 47(4):173–180, 1993.