Proceedings of SAT Race 2019: Solver and Benchmark Descriptions, volume B-2019-1 of Department of Computer Science Series of Publications B, University of Helsinki 2019.

Maple_LCM_OnlineDel

Sima Jamali
Simon Fraser University
Vancouver, Canada
sima_jamali@sfu.ca

David Mitchell
Simon Fraser University
Vancouver, Canada
mitchell@cs.sfu.ca

***Abstract*—We provide a brief introduction to our solvers *Maple_LCM_OnlineDel_19a* and *Maple_CM_OnlineDel_19b*. They use a simple online clause reduction scheme, which involves no sorting and was introduced in [10], within *Maple_LCM_Dist_ChronoBT*, the first place solver in the SAT Competition 2018 [1], [2].**

I. Introduction

Maple_LCM_Dist_ChronoBT won the gold medal of the main track of the SAT competition 2018 [3]. Its clause maintenance strategy, which was inherited from COMiniSatPS, involves three stores of learnt clauses: Core, Tier2 and Local [4], [5]. The size of Core and Tier2 are limited by being selective about which clauses are added. The majority of learnt clauses are added to Local. The size of Local is limited by periodically deleting half of the clauses with lowest activities, as in most recent CDCL solvers [4], [6], [7].

II. Online Deletion

The solvers described here replace the delete-half clause deletion strategy by a new deletion scheme called Online Deletion [10]. In Online Deletion, each time the solver derives a new conflict clause we choose a previously learnt clause to replace with it, according to the following simple method.

Clauses of Local are maintained in a circular list L with an index variable i that traverses the list in one direction. The index indicates the current "deletion candidate" L_i at each time. For deletion, we maintain a clause quality measure Q, and a threshold quality value q. When a new clause C is learnt and needs to be stored in Local, we select the next "low quality" clause in the list to be replaced with C. The index i is showing the next candidate L_i. While $Q(L_i) \geq q$, we increment i to "save" clause L_i for one more "round" in the circular list; The first time $Q(L_i) < q$, we replace L_i with C and delete the "old" clause L_i. The size of Local, indicates how long a round is, and the clause quality measure threshold is chosen in a way that there are always sufficiently enough "low-quality" clauses in the list to be deleted [10].

The submitted solvers use a simple quality measure based on counting how many times a clause has been used in conflict analysis since the last time it was considered for deletion. Q is calculated as follows:
Each clause has a quality measure RUL which is an indicator of its recent usage and LBD. RUL is initialized with 0 when a clause is first learnt. Every time a clause C is used in conflict analysis, its RUL is increase by $12/LBD(C)$. In the solvers submitted to this competition, the clause L_i is saved if its RUL is at least 2. ($Q(L_i) = RUL(L_i)$ and $q = 2$). If a clause is saved, its RUL resets to 0.

III. Maple_LCM_OnlineDel_19a

This solver is built based on the winning solver of the SAT competition 2018, Maple_LCM_Dist_ChronoBT [1]. The delete half clause deletion scheme is replaced by Online Deletion as described above. The maximum size of Local is set to 80,000, which indicates the "length" of each round during most of the run. The solvers was first introduced in [10] with minor difference in RUL (replaced $20/LBD(C)$ with $12/LBD(C)$).

IV. Maple_CM_OnlineDel_19b

This solver is the same as *Maple_LCM_OnlineDel_19a* with 2 differences:

1) Size of Local is set to 50,000
2) Unlike Maple_LCM_Dist_ChronoBT that uses the learnt clause minimization introduced in [8]. This solver has further learnt clause minimization as in the solver *Maple_CM_Dist*, which won the third place in the main track of SAT competition 2018 [9].

References

[1] A. Nadel and R. Vadim, "Chronological Backtracking," in Proceedings of SAT, 2018, pp. 111121.
[2] A. Nadel and R. Vadim, "Maple_LCM_Dist_ChronoBT: Featuring Chronological Backtracking," in Proceedings SAT Competition 2018 - Solver and Benchmark Descriptions, 2018, pp. 29.
[3] SAT Competition 2018, *http://sat2018.forsyte.tuwien.ac.at/index.php.*
[4] C. Oh, "Between SAT and UNSAT: The Fundamental Difference in CDCL SAT," in Proceedings of SAT, 2015, pp. 307323.
[5] C. Oh, "Improving SAT solvers by exploiting empirical characteristics of CDCL", Ph.D. dissertation, New York University, 2016.
[6] J. H. Liang, V. Ganesh, P. Poupart, and K. Czarnecki, Learning rate based branching heuristic for SAT solvers, in Proceedings of SAT,2016, pp. 123140.
[7] G. Audemard and L. Simon, Predicting learnt clauses quality in modern SAT solvers, in Proceedings of IJCAI, 2009, pp. 399404.
[8] M. Luo, C.-M. Li, F. Xiao, F. Manya, and Z. Lu, An effective learnt clause minimization approach for CDCL SAT solvers, in Proceedings of IJCAI, 2017, pp. 703711.
[9] M. Lou, F. Xiao, C. Li, F. Manya, Z. Lu and Y. Li, "Maple CM, Maple CM Dist, Maple CM ordUIP and Maple CM ordUIP+ ," in Proceedings SAT Competition 2018 - Solver and Benchmark Descriptions, 2018, pp. 4446.
[10] S. Jamali and D. Mitchell, "Simplifying CDCL Clause Database Reduction," in Proceedings of SAT, 2019.