Loandra, : Core-Boosted Linear Search for incomplete MaxSAT

Berg, Jeremias

2019

þÿ B e r g , J , S t u c k e y , P & E m i r D e m i r o v i , E , L o a n d r a , C o r e - B o o s t e d L i
incomplete MaxSAT , 2019 , Software .

http://hdl.handle.net/10138/315958

unspecified
acceptedVersion

Loandra: Core-Boosted Linear Search Entering MaxSAT Evaluation 2019

Jeremias Berg*, Emir Demirović†, Peter Stuckey†‡
*HIIT, Department of Computer Science, University of Helsinki, Finland
†University of Melbourne, Australia
‡Data61, CSIRO, Australia

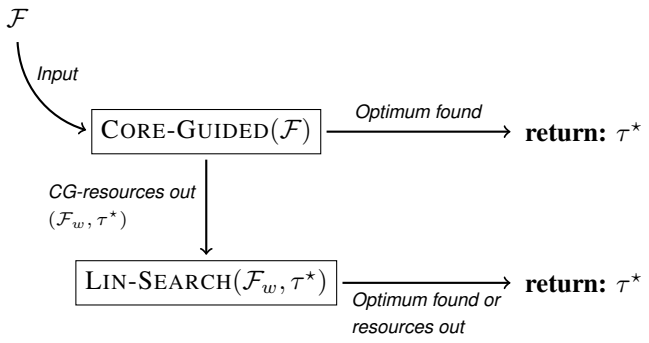

Fig. 1: The structure of Loandra.

I. Preliminaries

We briefly overview the Loandra MaxSAT-solver as it participated in the incomplete track of the 2019 MaxSAT Evaluation. A more thorough discussion can be found in [4]. We assume familiarity with conjunctive normal form (CNF) formulas and weighted partial maximum satisfiability (MaxSAT). Treating a CNF formula as a set of clauses a MaxSAT instance $\mathcal{F}$ consists of two CNF formulas, the hard clauses F_h and the soft clauses F_s, as well a weight w_c associated with each $C \in F_s$. A solution to $\mathcal{F}$ is an assignment τ that satisfies F_h. The cost of a solution τ is the sum of weights of the soft clauses falsified by τ. An optimal solution is one with minimum cost over all solutions. An unsatisfiable core κ of $\mathcal{F}$ is a subset of soft clauses s.t. $F_h \wedge \kappa$ is unsatisfiable.

II. Core-Boosted Linear Search

Loandra consists of two main components: CORE-GUIDED, a core-guided reformulation algorithm extended with stratification [1], and LIN-SEARCH, a SAT/UNSAT linear search algorithm. Both components make extensive use of Boolean Satisfiability (SAT) solvers. On input $\mathcal{F}$, CORE-GUIDED searches for an optimal MaxSAT solution by iteratively extracting unsatisfiable cores with a SAT solver and modifying a working instance (initialised to $\mathcal{F}$) in order to rule out them as sources of unsatisfiability. LIN-SEARCH iteratively queries the SAT solver for a solution of lower cost that the currently best known one. Both components are *complete*, i.e. given enough time and memory, both will compute an optimal solution. More importantly for the incomplete track, they also are *any-time*, i.e. both can output intermediate solutions during search. Note that stratification allows treating CORE-GUIDED as an any-time algorithm.

Figure 1 overviews the structure of Loandra. It uses core-boosting as described in [4] in order to exploit the strengths and alleviate the weaknesses of its individual components. On input $\mathcal{F}$ Loandra starts in a core-guided phase by invoking CORE-GUIDED on $\mathcal{F}$. If the optimal solution isn't found within the time allocated to the core-guided phase, the execution switches to a linear phase and LIN-SEARCH is invoked with $\tau^\star$, the best solution found by the core-guided phase, and $\mathcal{F}_w$, the final working instance of CORE-GUIDED. The linear search runs until either finding the optimal solution or reaching the time out, at which point the currently best known solution is returned.

III. Implementation details

The version of Loandra that entered the 2019 Evaluation is the same one as was experimented on in [4]. As the instantiation of CORE-GUIDED we use a reimplementation of the PMRES [8] MaxSAT algorithm extended with weight aware core extraction (WCE) [5] and clause hardening. The core-guided phase runs until no more cores can be found with the stratification bound set to 1, or 30s has passed.

As the instantiation of LIN-SEARCH we use a reimplementation of LinSBPS [3], SAT/UNSAT linear search extended with solution-based phase saving and varying resolution. Following LinSBPS we use the generalized totalizer encoding [6] to convert the PB constraints needed in linear search to CNF. All algorithms are implemented on top of the publicly available Open-WBO system [7] using Glucose 4.1 [2] as the back-end SAT solver.

IV. Compilation and Usage

Building and using Loandra resembles building and using Open-WBO. A statically linked version of Loandra in release mode can be built by running MAKE RS in the base folder.

After building, Loandra can be invoked from the terminal. Except for the formula file, Loandra accepts a number of command line arguments: the flag *-pmreslin-cglim* sets the maximum time that the core-guided phase can run for (in seconds). The rest of the flags resemble the flags accepted by Open-WBO; invoke ./loandra_static –help-verb for more information.

REFERENCES

[1] C. Ansótegui, M. L. Bonet, J. Gabàs, and J. Levy, "Improving SAT-based weighted MaxSAT solvers," in *Proc. CP*, ser. Lecture Notes in Computer Science, vol. 7514. Springer, 2012, pp. 86–101.

[2] G. Audemard and L. Simon, "Predicting learnt clauses quality in modern sat solvers," in *Proc IJCAI*. Morgan Kaufmann Publishers Inc., 2009, pp. 399–404.

[3] F. Bacchus, M. Järvisalo, R. Martins *et al.*, "Maxsat evaluation 2018," https://maxsat-evaluations.github.io/2018/, 2018, accessed: 2018-9-05.

[4] J. Berg, E. Demirovic, and P. Stuckey, "Core-boosted linear search for incomplete maxsat," in *Proc CPAIOR*, ser. Lecture Notes in Computer Science, vol. ???? Springer, 2019, p. ??? (to appear).

[5] J. Berg and M. Järvisalo, "Weight-aware core extraction in SAT-based MaxSAT solving," in *Proc. CP*, ser. Lecture Notes in Computer Science, 2017, to appear.

[6] S. Joshi, R. Martins, and V. M. Manquinho, "Generalized totalizer encoding for pseudo-boolean constraints," in *Proc. CP*, ser. LNCS, vol. 9255, 2015, pp. 200–209.

[7] R. Martins, V. Manquinho, and I. Lynce, "Open-WBO: A modular MaxSAT solver," in *Proc. SAT*, ser. Lecture Notes in Computer Science, vol. 8561. Springer, 2014, pp. 438–445.

[8] N. Narodytska and F. Bacchus, "Maximum satisfiability using core-guided MaxSAT resolution," in *Proc. AAAI*. AAAI Press, 2014, pp. 2717–2723.